

# **HAL Software Package**

User manual for BeamConstruct  
and additional tools

Version 7.7

(c) 2008-2024 by HALaser Systems GmbH

# Table Of Contents

1 Copyright.....	5
2 Legal Issues.....	6
3 Safety.....	10
4 Overview.....	11
4.1 The Software Structure.....	11
5 BeamConstruct.....	12
5.1 Security.....	12
5.2 Overview.....	12
5.3 Quick Start into BeamConstruct.....	12
5.4 User interface.....	15
5.4.1 The Project menu.....	15
5.4.2 The Help menu.....	16
5.5 Project Configuration.....	18
5.5.1 Scanner Controller Configuration.....	25
5.5.2 Motion Controller Configuration.....	56
5.5.3 Image Capture Configuration.....	62
5.5.4 Additional Laser Control Configuration.....	63
5.5.5 Z-Shifter Configuration.....	65
5.5.6 Process Controller Configuration.....	67
5.6 Machine Configuration.....	71
5.6.1 Preview Projector.....	73
5.7 Pen Settings.....	74
5.7.1 General Pen Settings.....	75
5.7.1.1 Pen Parameter Wizard.....	76
5.7.2 Scanner Pen Settings.....	76
5.7.2.1 Sky-Writing.....	82
5.7.2.2 Overspeed.....	84
5.7.3 Bitmap Pen Settings.....	85
5.7.4 Laser Pen Parameters.....	85
5.7.5 Using Pens for Coloured Bitmap Marking.....	86
5.8 Construction Of Geometry.....	87
5.8.1 The Drawing Area.....	88
5.8.1.1 Select Elements.....	89
5.8.1.2 2D and 3D Editing.....	90
5.8.1.3 Drawing Area with 3D surface.....	90
5.8.2 The Property Tab-Panes.....	91
5.8.2.1 The tab-pane Of The Currently Selected Element.....	91
5.8.2.2 The Geometry Tab-Pane.....	91
5.8.2.3 The Element Tab-Pane.....	92
5.8.2.4 The Layer Tab-Pane.....	94
5.8.2.5 The Motion Tab-Pane.....	94
5.8.3 The Element-Tree.....	95
5.8.3.1 Auto-Resorting of elements in groups.....	96
5.8.3.2 Auto-Arrangement of elements in groups.....	97
5.8.4 Primary Geometry Elements.....	97
5.8.4.1 Dot.....	97
5.8.4.2 Line.....	98
5.8.4.3 Triangle.....	98
5.8.4.4 Rectangle.....	99
5.8.4.5 Circle.....	99
5.8.4.6 Star.....	100

5.8.4.7	Spiral.....	101
5.8.4.8	Polygon.....	102
5.8.4.9	Bezier Curve.....	102
5.8.4.10	Text.....	103
5.8.4.11	Barcode.....	105
5.8.4.12	Delay.....	106
5.8.4.13	Motion.....	106
5.8.4.14	Custom Output.....	107
5.8.4.15	Custom Input.....	108
5.8.4.16	Program Flow.....	109
5.8.4.17	External Trigger.....	109
5.8.4.18	Serial port output.....	110
5.8.4.19	Z-Shifter.....	110
5.8.5	Additional Geometry Elements.....	110
5.8.5.1	Hatcher.....	111
5.8.6	Post-processing Tool Elements.....	115
5.8.6.1	Sine Distortion.....	115
5.8.6.2	Curve Distortion.....	115
5.8.7	Data Input Elements.....	116
5.8.7.1	CSV Data Input.....	116
5.8.7.2	OpenAPC Data Input.....	117
5.8.7.3	Serial/Date/Time.....	117
5.9	Edit Geometry.....	119
5.9.1	Edit Vectors.....	119
5.9.2	Edit Non-Destructive.....	120
5.9.2.1	Active Split Group.....	120
5.9.2.2	Active Move Group.....	122
5.9.3	Edit Destructive.....	123
5.10	Process 3D Data.....	125
5.10.1	Slicing 3D Data.....	125
5.10.2	Sliced 3D Mesh Group.....	126
5.10.3	3D Mesh.....	131
5.10.4	3D Slice Cleaner.....	132
5.10.5	Auto-Arrangement of 3D Meshes in Sliced 3D Mesh Groups.....	132
5.11	Vision.....	133
5.11.1	Vision Functions.....	133
5.11.2	Camera Calibration Cookbook.....	134
5.11.3	Fiducial Calibration Cookbook.....	136
5.12	Processing Operation.....	136
5.13	Marking.....	137
5.14	Switching the Laser.....	140
5.15	Exporting Data.....	140
5.16	Multihead Operations.....	141
5.17	IOSelect Mode.....	143
5.18	Command line options.....	144
5.19	Customise BeamConstruct.....	145
5.19.1	Rebranding.....	145
5.19.2	In-application images.....	145
5.19.3	Custom menus and functions.....	147
5.19.4	Predefined hardware settings.....	149
5.19.5	Custom geometries.....	149
5.19.6	Adding Languages.....	150
5.19.6.1	Working Principle.....	150
5.19.6.2	Choosing the Correct Translation File Name.....	150

5.19.6.3 Translating the Applications.....	151
5.20 BeamLock.....	151
5.20.1 Fixed time range locks.....	152
5.20.2 Dynamic time range lock.....	153
5.21 BeamServer Remote Control Interface.....	153
5.21.1 Overview.....	153
5.21.2 Usage.....	153
5.21.3 Starting BeamServer.....	154
5.21.4 Remote Control Commands.....	155
5.21.4.1 Element Flags.....	178
5.21.4.1.1 Scanner Bitmap Primary Element Flags.....	178
5.21.4.1.2 Z-Shifter Primary Element Flags.....	179
5.21.4.1.3 Motion Primary Element Flags.....	179
5.21.4.1.4 Spiral Primary Element Flags.....	180
5.21.4.1.5 Circle Primary Element Flags.....	180
5.21.4.1.6 Custom Output Primary Element Flags.....	180
5.21.4.1.7 Barcode Primary Element Flags.....	180
5.21.4.1.8 Serial Number Input Element Flags.....	181
5.21.4.1.9 Curve Postprocessing Element Flags.....	181
5.22 Smart Interface.....	181
5.22.1 Accessing the Smart Interface.....	181
5.23 MQTT Data Interface.....	185
5.23.1 MQTT Messages.....	186
6 CorrCorrect.....	188
6.1 Overview.....	188
6.2 Usage.....	188
6.3 Correction Calculation Dialogue.....	189
6.3.1 Angle Calculation Wizard.....	189
6.4 Correction Interpolation Dialogue.....	190
6.5 Correct Single and Multiple Points Dialogue.....	191
6.6 Spatial Correction Dialogue.....	191
6.7 Rising Scale Correction.....	192
6.8 Import Scanned Grid.....	192
6.9 CorrCorrect Cookbook.....	194
7 Firmware Downloader.....	197
8 BeamControl.....	198

# 1 Copyright

This document is © by HALaser Systems. OpenAPC, ControlRoom and BeamConstruct are registered trademarks.

Other software described here that not belong to the HAL software package is © and TM by their respective owners.

Fedora, RedHat, RedHat Enterprise Linux, RHEL are copyright / trademarks / legal trademarks of Red Hat Inc.

Linux is a trademark / legal trademark of Linus Torvalds.

Ubuntu is copyright / trademark / legal trademark of Canonical.

Microsoft, Windows, the Windows-logo are copyright / trademarks / legal trademarks of Microsoft Corporation.

Panasonic and Minas are copyright / trademarks / legal trademarks of Matsushita.

MDrive and MDrive+ are copyright / trademarks / legal trademarks of Schneider Electronic.

Isel and Isel Wafer Handler Robot are copyright / trademarks / legal trademarks of Isel.

JoyWarrior is copyright / trademark / legal trademark of Code Mercenaries Hard- and Software GmbH.

SCAPS, SAM, CCI, FEB and USC are copyright / trademarks / legal trademarks of SCAPS GmbH.

SCANLAB, RTC, RTC3, RTC4, RTC5, RTC6, RTCscanalone and others are copyright / trademarks / legal trademarks of SCANLAB AG.

SiRF is copyright / trademark / legal trademark of SiRF Technology Inc.

Weecoboard, Weecoboard 4M and Aptasys is copyright / trademark / legal trademark of Aptasys s.r.l.

SP-ICE2, RLC, Raylase and others are copyright / trademarks / legal trademarks of Raylase AG.

ETH6608, Sintec and others are copyright / trademarks / legal trademarks of Sintec Optronics.

PSC, PSC140P, 3rdEye and others are copyright / trademarks / legal trademarks of 3rdEye.

Coherent, Avia and others are copyright / trademarks / legal trademarks of Coherent Inc.

SPI and others are copyright / trademarks / legal trademarks of SPI Lasers Ltd.

Sill and others are copyright / trademarks / legal trademarks of Sill Optics GmbH & Co. KG.

Hermes and The Hermes Standard are copyright / trademarks / legal trademarks of the Hermes standardisation consortium.

All other trademarks mentioned in this document are trademarks or registered trademarks by their respective owners.

All rights not expressly granted herein are reserved.

## 2 Legal Issues

### LICENSE AGREEMENT

The HAL software package that is made available (the "Software") is owned by HALaser Systems (the "Vendor"). This Software is protected by copyright laws, and is being made available solely for use by you in accordance with the following terms and conditions. Any use, reproduction or redistribution of the Software that is not in accordance with this Software License Agreement is expressly prohibited by law, and may result in civil and criminal penalties.

### SOFTWARE LICENSE AGREEMENT

THE VENDOR IS WILLING TO LICENSE THIS SOFTWARE TO YOU ONLY ON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS LICENSE AGREEMENT. This is a legal agreement between you (either an individual end-user or an entity) and the Vendor ("Agreement"). By using this software, you are agreeing to be bound by the terms and conditions of this Agreement. If you do not agree to the terms and conditions of this Agreement, promptly return the software and other items that are part of this product in their original package with your sales receipt to your point of purchase for a full refund, or if you have downloaded this software from the Vendors web site, then you must stop using the software and destroy any copies of the software in your possession or control.

#### 1. Grant of Agreement.

Subject to the terms and conditions of this Agreement, the Vendor and its suppliers grant to you a non-exclusive license to use one licensed copy of the Software and any documentation accompanying this Agreement on one computer. No other rights are granted. The Software is in use if it is loaded on the computer's permanent or temporary memory. For backup purposes only, you may make one copy of the Software. You must include on the backup copy all copyright and other notices included on the Software as supplied by the Vendor. Installation on a network server for the sole purpose of your internal distribution of the Software is permitted only if you have purchased an individual Software license for each networked computer to which the Software is distributed.

#### 2. Restrictions.

The Software may contain copyrighted material, trade secrets, and other proprietary materials of the Vendor and its licensor's. You agree that in order to protect those proprietary materials, except as expressly permitted by applicable law, neither you nor a third party acting on your behalf will:

- (I) decompile, disassemble or reverse engineer the Software;
- (II) modify or create derivative works of the Software;
- (III) transmit the Software, in whole or in part, over the Internet or other network; or
- (IV) sell, distribute, rent, lease, sub license or otherwise transfer the Software to a third party; that does not influence your right to sell the Software together with additional equipment that is controlled by this Software

#### 3. Ownership.

The Software is licensed, not sold, to you for use only under the terms and conditions of this Agreement, and the Vendor reserves all rights not expressly granted to you in this Agreement. The Vendor and/or its licensor's retain title to the Software, and all intellectual property rights therein.

#### 4. Termination.

This Agreement is effective until terminated. Upon any violation of any of the provisions of this Agreement, rights to use the Software shall automatically terminate and the Software must be returned to the Vendor or all copies of the Software must be destroyed. You may also terminate this Agreement at any time by destroying all copies of the Software in your possession or control. If the Vendor makes a request via public announcement or press release to stop using the copies of the Software, you will comply immediately with this request. The provisions of paragraphs 3, 7, 8 and 12 will survive any termination of this Agreement.

## 5. Limited Product Warranty.

The Vendor warrants to you that the Software will substantially conform to its published documentation and the media containing the Software shall be free from defects in material, each for a period of thirty (30) days from the date of purchase. The Vendors limited warranty is non-transferable and is limited to the original purchaser. This warranty gives you specific legal rights, and you may also have other rights which vary under local laws.

## 6. Remedies.

The Vendors entire liability and your exclusive remedy for any breach of warranty shall be, at the Vendors option, to:

- (a) repair or replace the Software or media, provided that the Software or media is returned to the point of purchase or such other place as the Vendor may direct, with a copy of the sales receipt, or
- (b) refund the price paid.

Any replacement Software or media will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer. These remedies are void if failure of the Software or media has resulted from accident, abuse, or misapplication.

## 7. DISCLAIMER OF WARRANTY.

THE WARRANTIES EXPRESSLY SET FORTH IN THIS AGREEMENT REPLACE ALL OTHER WARRANTIES. THE VENDOR AND ITS SUPPLIERS EXPRESSLY DISCLAIM ALL OTHER WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD-PARTY RIGHTS WITH RESPECT TO THE SOFTWARE OR MEDIA, AND ANY WARRANTIES OF NON-INTERFERENCE OR ACCURACY OF INFORMATIONAL CONTENT. NO DEALER, AGENT, OR EMPLOYEE OF THE VENDOR IS AUTHORIZED TO MAKE ANY MODIFICATION, EXTENSION, OR ADDITION TO THIS WARRANTY.

Some jurisdictions do not allow limitations on how long an implied warranty lasts, so the above limitation may not apply to you.

## 8. LIMITATION OF LIABILITY.

IN NO EVENT WILL THE VENDOR OR ITS SUPPLIERS BE LIABLE FOR ANY COSTS OF PROCUREMENT OF SUBSTITUTE PRODUCTS OR SERVICES, LOST PROFITS, LOSS OF INFORMATION OR DATA, OR ANY OTHER SPECIAL, INDIRECT, CONSEQUENTIAL, OR INCIDENTAL DAMAGES ARISING IN ANY WAY OUT OF THE SALE OF, USE OF, OR INABILITY TO USE ANY VENDORS PRODUCT OR SERVICE, EVEN IF THE VENDOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO CASE SHALL THE VENDOR ITS SUPPLIERS' TOTAL LIABILITY EXCEED THE ACTUAL MONEY PAID FOR THE VENDORS PRODUCT OR SERVICE GIVING RISE TO THE LIABILITY.

Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you. The above limitations will not apply in case of personal injury where and to the extent that applicable law requires such liability.

## 9. Export Law Assurances.

You agree and certify that neither the Software nor any other technical data received from the Vendor will be exported outside the Federal Republic of Germany except as authorized and as permitted by the laws and regulations of the Federal Republic of Germany. If you have rightfully obtained the software outside of the Federal Republic of Germany, you agree that you will not re-export the Software nor any other technical data received from the Vendor, except as permitted by the laws and regulations of the Federal Republic of Germany and the laws and regulations of the jurisdiction in which you obtained the Software.

## 10. Agents and Third Party Purchasers.

If you are acquiring the Software on behalf of another person or entity, you represent and warrant that you have the authority to bind the party or entity for which you are acquiring the Software to the terms and conditions of this Agreement.

## 11. General Terms and Conditions.

This Agreement will be governed by and construed in accordance with the laws of the Federal Republic of Germany, without regard to or application of its choice of law rules or principles. If for any reason a court of competent jurisdiction finds any provision of this Agreement, or portion thereof, to be unenforceable, that provision of the Agreement shall be enforced to the maximum extent permissible so as to affect the intent of the parties, and the remainder of this Agreement shall continue in full force and effect. This Agreement constitutes the entire agreement between the parties with respect to the use of the Software and supersedes all prior or contemporaneous understandings, communications or agreements, written or oral, regarding such subject matter. The Vendor may, in its sole discretion, modify portions of this Agreement at any time. The Vendor may notify you of any changes by posting notice of such modifications on the Vendors web site(s) or sending notice via e-mail, postal mail or other means. Your continued use of the Software following notice of such modifications shall be deemed to be your acceptance of any such modifications to the Agreement. If you do not agree to any such modifications, you must immediately stop using the Software and destroy all copies of the Software in your possession or control.

12. The Software is protected by Federal Republic of Germany copyright law and international treaty. Unauthorized reproduction or distribution of the Software is subject to civil and criminal penalties.

### NOTICE SPECIFIC TO INFORMATION AVAILABLE ON THE WEBSITE OF THE VENDOR:

The information provided is for informational purposes only and is subject to change without notice. It is provided "AS IS" and without any warranty. Any risk arising out of the use of the information on this website shall remain with the reader. IN NO EVENT SHALL THE VENDOR BE LIABLE FOR ANY INDIRECT, CONSEQUENTIAL, INCIDENTAL, SPECIAL, PUNITIVE OR OTHER DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION OR LOSS OF BUSINESS INFORMATION), ARISING OUT OF OR IN CONNECTION WITH THE AVAILABILITY, PERFORMANCE OR USE OF THE INFORMATION CONTAINED ON THIS WEBSITE, EVEN IF THE VENDOR HAS BEEN ADVISED OF THE POSSIBILITIES OF SUCH DAMAGES.

### LINKS TO OTHER WEB SITES

Links to other web sites are available on the Vendors web site and these links allow you to leave the Vendors web site. The linked sites are not under the control of the Vendor, the Vendor is not responsible for the contents of any linked site or links within a linked site. These links are provided as a convenience only and do not imply any endorsement or recommendation by the Vendor.

### PORTIONS OF THIS SOFTWARE CONTAIN THE ASSIMP LIBRARY v4.0 OR LATER:

Open Asset Import Library (assimp)

Copyright (c) 2006-2020, assimp team  
All rights reserved.

Redistribution and use of this software in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the assimp team, nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission of the assimp team.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR



ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

\*\*\*\*\*

AN EXCEPTION applies to all files in the ./test/models-nonbsd folder. These are 3d models for testing purposes, from various free sources on the internet. They are - unless otherwise stated - copyright of their respective creators, which may impose additional requirements on the use of their work. For any of these models, see <model-name>.source.txt for more legal information. Contact us if you are a copyright holder and believe that we credited you improperly or if you don't want your files to appear in the repository.

\*\*\*\*\*

Poly2Tri Copyright (c) 2009-2010, Poly2Tri Contributors  
<http://code.google.com/p/poly2tri/>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Poly2Tri nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### **3 Safety**

The HAL software package components are designed to control different kind of machinery. This machinery may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured. The client shall solely be responsible to strictly comply with all applicable and relevant safety regulations regarding installation and operation of the system at any time.

## 4 Overview

This document describes the full HAL software package including all existing functionalities, options and additional software components. Depending on the software variant you really use some of the described features and functionalities may be not available or may be available only in a limited state.

The HAL software package is available for different hardware and software platforms. Dependent on the peculiarities of the used operating systems the exact storage position of the executable, the plug-ins and other required data may differ, here always a storage location is chosen that is a common position for such a software on that operating system. Within this document no operating system specific information are given, here please refer to the user manual of the operating system.

### 4.1 The Software Structure

The HAL software package is a collection of software components and additional tools that can be used for different purposes. The functionality of all of them can be combined in different ways depending on the type of application.

## 5 BeamConstruct

### 5.1 Security

This application is designed to control laser equipment which may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured.

Beside of that some laser equipment can be damaged in case it is controlled with wrong signals. Thus it is highly recommended to check the output generated by this plug in using e.g. an oscilloscope to avoid problems caused by wrong configurations. This should be done prior to putting a system into operation for the first time and whenever a software update was installed or whenever some of the relevant parameters have been modified.

Here “software update” not only means updates of the HAL package or BeamConstruct software but also modifications with dependent software, like components or drivers for scanner controller cards.

### 5.2 Overview

BeamConstruct is a CAD application specialised for creation and processing of laser marking data. These data can be used together with a laser scanner system to perform the material processing.

### 5.3 Quick Start into BeamConstruct

This section describes how to use BeamConstruct in direct operation mode where laser marking is done directly out of the application. It gives a short overview about basic set-up and usage of BeamConstruct and describes with some short steps how to get first results easily. So this section is a shortcut to start into this software very fast but it ignores most of the useful features. These features are described in detail in following sections of the users manual, so it is recommended to use this quick start only to understand the very basic working principles of the software while going deeper into it afterwards.

The very first and most important rule, when starting to work with BeamConstruct is: **DON'T TOUCH!** BeamConstruct comes with default settings that all make sense for a common, non-complex setup and therefore can be used quite well for the beginning. So unless you exactly know what you are doing and why you are changing some values, don't modify any parameters. At the beginning, when you do your first steps with BeamConstruct, try to work with these default parameters until you stumble upon a problem, where you explicitly have to change some specific parameters to solve this problem.

1. **SECURITY CHECK:** The following steps describe how to set up BeamConstruct and how to control laser equipment out of it. Thus all laser safety rules and regulations need to be respected, all required technical security mechanisms need to be available and active prior to starting with this software.
2. **Main configuration:** First the scanner controller card that has to be used needs to be selected and configured. To do that, select Menu “Project” menu item “Project settings...” and go to panel “Hardware” of the now opening settings dialogue. Within the selection-list named with “Scanner Card” choose the scanner controller card you want to use. After selecting it, press the button “Configure” and set up the scanner controller card according to its specific needs (choose things like firmware file, correction table, working area size, number of controlled axes or what ever your card requires for proper operation) and according to your local hardware configuration.
3. **Save Configuration:** Leave all these settings dialogues by pressing “OK” once all parameters have been entered correctly and select menu “Project” menu item “Save as default configuration”. Now

whenever you start BeamConstruct and as long as you do not load a project file with a different hardware set-up, these settings are used by the application.

4. **Create basic geometry:** Select the blue triangle-symbol within the tool bar and draw that triangle within the drawing area right below the tool bar. There you have to left-click your mouse three times at different positions to create the triangle, every of these mouse-clicks specifies an other corner of the triangle.
5. **Modify existing geometry interactively:** Once you have finished creation of the triangle this new element is selected and highlighted by a double-lined blue box surrounding it. This box can be used to modify the geometry. While picking and dragging one of the coloured squares that are contained in this surrounding, the triangles size can be changed (the grey squares) as well as its rotation (via the red square). The cyan-coloured squares can be used to slant the geometry. When the triangle is dragged by clicking the selection frame between these coloured squares, its position can be changed.  
PLEASE NOTE: When modification of the geometry has finished it still has to be located completely within the working area that is symbolised by the grey rectangle. This working area is the maximum range your scanner can work within, so all geometry that is located outside of that range will be cropped.
6. **Modifying existing geometry manually:** The method described previously is a very quick way to change the geometry but it is also a very inaccurate one. Thus there exists an other possibility: As long as an element is selected, there is a configuration panel usable at the windows left hand side. The first configuration panel is always an element-specific one, in this example it contains several parameters and values that influence the creation of the triangles base geometry. On the same side of the main window there also exists an other tab-pane "Geometry" where the generic geometric data of an element can be changed by entering the desired numeric values. Using this way things like size, position and rotation can be modified in a very exact way. The related values are taken over and applied to the geometry as soon as the data input is confirmed by pressing "Return" or by leaving the input field.  
Several of these data depend on each other and will change automatically whenever one of them is modified. So in case you change the scale factor of your geometry, the position and size values of it will be changed too as soon as you hit your "Return"-key within the scale value input field.
7. **Apply laser parameters:** Required laser parameters can be applied to existing geometry via a number of predefined and freely definable pens. Such an (existing) pen can be set via the tab-pane "Element" and the selection list named "Pen". Whenever a new pen is selected for an existing geometry it is applied to it automatically, which means the visual representation of that geometry within the drawing area changes its colour and the next laser marking cycle would use the parameters of this pen when this specific geometry is processed.  
The different laser and scanner parameters that are assigned to such a pen can be modified by selecting menu "Project" menu item "Pen settings...". There a dialogue opens where you can select the pen that has to be modified (by using the selection list on the upper side of the dialogue). The tab-panes below this list give access to all relevant parameters that influence mark and jump speed, laser power and frequency, scanner delays and others more. When you change some of these parameters and leave the dialogue by pressing "OK" all the geometries that are assigned to this (now changed) pen will use the modified parameters during next laser marking process.
8. **Adding a hatch pattern:** Currently only the outline of the triangle is shown and only this outline would be processed during laser marking. To process the inner side of the triangle too, a fill pattern has to be added by using the Additional Geometry element "Hatch". It is symbolised by a purple-coloured tool bar icon that consists of several horizontal lines. When this icon is clicked as long as a Primary Geometry element (like the triangle) is selected, it is added to this element automatically. Now the appearance of the geometry and the appearance of the application changes:
  - the first panel on the main windows left hand side switches to show all the "Hatch" parameters
  - the Element Tree on the main windows right hand side now shows the new hierarchical structure of the element with the "Hatch" as sub-element of the primary "Triangle" geometry
  - the triangle within the drawing area is filled with some hatch linesThis hatch pattern now can be changed by modifying the values within the property-panel "Hatch" on the main windows left hand side.  
PLEASE NOTE: after clicking the Hatch-button within the tool bar, the new geometry of the now added hatch/fill pattern is selected instead of the triangle! That means all modifications to the selected geometry (as described previously) now would apply to the hatch geometry only but NOT to

its parents triangle! So in order to manipulate the whole element the parent of the hatch has to be selected – which is the base element generated in previous step.

9. **Save data:** To avoid the currently generated geometry gets lost, the project has to be saved now. To do that, select menu “Project” menu item “Save as...” and save these data at a suitable position using the .BEAMP format.  
PLEASE NOTE: this project contains all the hardware definitions that have been done during configuration of the software in first steps. Whenever you change this configuration and then load this project again, the hardware set-up of the project is used instead of the changed one of the application. To avoid that after loading such a project, the menu “Project” menu item “Load default configuration” can be chosen to replace the projects hardware set-up by the local, desired one.
10. **SECURITY CHECK:** Next the scanner controller card together with a possibly connected laser will be accessed for the first time. That means it is opened and initialised and all connected equipment may start working now. Thus it is very important to ensure all security regulations are met and nobody can be injured and no damage can be caused also in case laser output or other motion starts spontaneously and unexpectedly!
11. **Prepare for laser marking:** Now the created geometry has to be sent to the connected laser and scanner system for processing it on some material. To do that the menu “Process” menu item “Mark” has to be selected. It opens the marking dialogue and tries to access the connected hardware. In case it is not possible to access the scanner controller card, an error message is shown and none of the buttons of the mark dialogue are usable (except the “Cancel” button to leave the dialogue). In this case you have to go back to the project settings as described in step 2 and to correct the set-up of your scanner controller card.
12. **SECURITY CHECK:** Next the laser and scanner will be accessed. Since there are some (laser) controllers available that are VERY sensitive to wrong laser signals, it is recommended first to check the output your scanner controller card produces. This can be done e.g. by using an oscilloscope instead of the real hardware. The target equipment should be connected only in case all signals are checked, correct and acceptable by the hardware. More than this it is recommended to repeat this step whenever the something within the complete set-up changed; changes may be caused e.g. by modified parameters, by software or driver updates.
13. **Start marking:** Now marking can be started by pressing the big yellow button (the one with the laser warning sign) in the middle of the marking dialogue. This starts sending of all laser and scanner data to the scanner controller card so that it is able to output motion and laser control data synchronously. Such a marking operation can be stopped by pressing the big red button (with the STOP-symbol) at the dialogues right hand side.  
PLEASE NOTE: this stop-button is not a replacement for a real emergency stop, it will try to stop the currently running laser process only via software which – in worst case – may fail for an example due to an abnormal problem or in case of a hardware failure. Thus there must be independent and working emergency stop equipment available in every case!

## 5.4 User interface

### 5.4.1 The Project menu

The menu “Project” gives access to different global, project- or application-specific operations:

- New project – create a new project, this deletes all possibly existing data and sets BeamConstruct to default configuration values
- Open project – open an existing .BEAMP BeamConstruct project which contains all required data like hardware settings, geometries, control elements, pen settings,...; when using this function, all data of a previously used project will get lost
- Insert project – load a .BEAMP project without deleting the current one: the hardware settings of the current project are left unchanged and the elements and pen settings of the new project are appended to the new one
- Load hardware settings – loads hardware settings out of a .BEAMH file while leaving possibly existing elements (control and geometry) and pen settings unchanged
- Load payload data – loads geometries, control elements and pen settings out of a .BEAMV file while leaving only the hardware settings unchanged
- Import – load data out of a file that is not a BeamConstruct project file (.BEAMP or .BEAMV). The newly loaded geometries are added to the current project. Depending on the capabilities of the chosen import format, a dialogue is opened here which gives the possibility to define several import parameters. When a raster image (such as BMP, JPG, GIF, PNG,...) is imported, the user can choose between different options how to deal with the raster data:

Use as scanner bitmap	The picture is kept as raster image and marked as greyscale or b/w bitmap. Such an element always should make use of an own pen which uses bitmap-specific pen-settings (as described in section “5.7.3 Bitmap Pen Settings”)
Use as coloured scanner bitmap	Same as the standard bitmap as described above but supports marking of coloured images as described in section “5.7.5 Using Pens for Coloured Bitmap Marking”
Use as light-table image	When this option is chosen, the imported bitmap becomes part of the project but it is not marked. It is visible as some background image only and e.g. can be used as some template to draw vector geometries on top of it which itself then are marked
Point cloud	<p>Instead of keeping a raster image, when using this option, the image is converted into a point cloud which is similar to a dithered black/white image. Here instead of having single pixels within the bitmap, each dot is a real vector point. Following parameters can be chosen here:</p> <ul style="list-style-type: none"><li>• Assign to pen – the pen which has to be set for the points. The laser spot-size in this pen defines the size and therefore the number of points to be created and corresponds to the following parameter:</li><li>• Overlap dots – the ratio the (round) dots have to overlap each other independent from their original positions where they would not overlap each other</li><li>• Resulting size – size the point cloud has to cover after importing; here an option “Keep aspect ratio” can be chosen to allow a more convenient selection of a proper size</li><li>• Orientation – the orientation of the resulting point cloud can</li></ul>

be selected here

- Brightness layers: normally a raster image is converted to one point cloud where the dots in this cloud define pixels in the input image which are below of a specific brightness level. With the brightness-layer-options, it is possible to have a defined number of different point clouds which each represent a different brightness level of the pixels of the original image. Via the “Brightness layer distance” it is possible to place each of these layers to a different Z-level. The “Layer levels” define the threshold for each of the layers to be created, so they are responsible for sorting the points into the different cloud-layers depending on the brightness of their source pixel.

Beside of these import-specific options, in upper part of the import dialogue there are different parameters to correct the input image. A preview right beside these sliders shows what the result will look like when the image is imported with these correction values.

When a 2D import format is used which results in a marking geometry (but not in a 3D mesh), and when these data are imported into one single, non-grouped and non sub-structured element, an option “Reload on file change” is offered in the property tab-pane of this imported element. When this option is checked, now whenever the source-file on disk changes, the file is imported again. This function can be used to easily implement a simple reload-function which allows it to replace the currently loaded geometry quick and easy.

- Save project – save the currently loaded project using the same name, this operation overwrites an existing project without further notice
- Save project as – save the currently loaded project while giving it a (new) name by using a file dialogue
- Save with options – save the currently loaded project with the possibility to specify what this project may contain exactly
- Save hardware settings – save the current hardware settings into a .BEAMH file
- Save payload data – save geometries, control elements and pen settings of the current project into a .BEAMV file
- Export as – here data or parts of data of the currently loaded project can be exported using one of the following file formats:
  - CSV – saves the 2D vector data using comma separated values
  - HPGL – saves the 2D vector data using HPGL/PLT file format
  - CLI – saves the different slices of an already processed 3D mesh in CLI layer format
  - RWX – saves the model data of all 3D meshes of the current project in RWX 3D file format
- Project settings – open the global settings dialogue where all application and hardware related configurations can be done; for details please refer to section “5.5 Project Configuration”
- Pen settings – open the dialogue where all material and output related parameters can be configured; for details please refer to section “5.7 Pen Settings”
- Save as default configuration – save the current hardware and pen settings as default so that these parameters are loaded as default on next start-up of BeamConstruct or when a new project is created by selecting menu “New project”
- Load default configuration – when a process file was loaded, it possibly can contain an own hardware configuration which does not fit to the current device. Using this menu item the local default configuration can be loaded overriding the parameters of the loaded project

## 5.4.2 The Help menu

This menu gives access to different informational and service functions:



- About – displays information about software version, copyright and vendors homepage
- License – displays information about the current license, when your application pops up with an error message “Insufficient license” at some functions, please check here if you really have the license which is required
- Reset license – this function can be used to get rid of a software license which was retrieved online. After this function has been called, BeamConstruct is no longer licensed and runs in demo mode
- Generate bugreport – this function can be used in case of problems with the software to gather different data which are helpful to analyse these problems.  
Please note: this function collects different information which may include sensitive, personal or other confidential data. When you do not agree to share these data, do not use this function. Otherwise you agree that we get, store, analyse and read these data for an undefined time (but typically only as long as it is necessary to check your specific problem).

When you see a problem in the software, some kind of misbehaviour where application does not crash and still can be used, please perform the following steps:

- after the problem appeared, do not use any other functions, only in case some overlapping windows or dialogues are still open, please close them
- after that, directly go to menu "Help" → "Generate bugreport"
- let the software create the bugreport and send the resulting ZIP-file to us together with a description what you have done, what result you have expected and what result you have seen instead

When you experience a crash or a situation, where you have to kill the application, please perform the following steps to give us the possibility to analyse the problem:

- restart the software immediately
- load the latest project you have seen the crash or deadlock with
- go to menu "Help" → "Generate bugreport"
- let the software create the bugreport and send the resulting ZIP-file to us together with a description what you have done exactly to get this crash
- Save project protocol – saves several data of the current project and configuration into a CSV file. These data can be used e.g. during development to track changes or for documentation purposes while experimenting with different parameters and settings
- Credits – displays credits regarding to people and companies which contributed things to this software

## 5.5 Project Configuration

When starting with a new project it has to be configured first so that the editing environment of BeamConstruct reflects the real processing conditions within production.

This can be done in global settings, the related dialogue can be found in menu “Project” sub-menu “Project settings...”. There a dialogue opens which offers several configuration options within different tab panes. The following parameters can be set and modified within these panes:

In tab-pane “User interface”:

- Visual grid size – this is a helper grid that is used and visible within the working area as an array of points with a distance in X and Y direction as specified here
- Snap to Grid – when this check box is selected, all geometry that is drawn or re-positioned within the working area automatically snaps to a position that is defined by the visual grid; this option gives the possibility to position elements very exact
- Background Colour – this option changes the visual representation of the editor and does not have any influence on the generated data, it specifies the colour of the background within BeamConstruct's drawing area
- Working area Border – the working areas (as defined in scanner controller settings for every scanhead) are shown as rectangle within the drawing area of the application, here the colour of the bounds of related rectangles can be changed; similar to the previous option this one also does not influence the generated data
- Editing area border – beside the working area, which specifies the range a single scanhead can cover, an editing area exists which is a visual help showing the user where drawing is allowed for the current configuration. This area exists only once for a configuration, it can be equal or smaller than the working area of a single scanhead (in single head environments), it can include several scanheads working areas (in multihead environments) or it can be larger than the available scanheads working area (e.g. when it has to be used together with the functionality of Active Split Group). Here the colour of the bounds of this area can be changed; similar to the previous option this one also does not influence the generated data
- 3D mesh colour – this colour is used in 3D view only and specifies how an imported 3D model is drawn
- 3D support mesh colour – this colour is used in 3D view only and specifies how the support structures for a sliced 3D model are drawn
- Editing area upper left – specifies the upper left corner of the global editing area within the coordinate system of the application
- Editing area size – specifies the size of the global editing area within the coordinate system of the application

In tab-pane “Scanner” it is possible to configure the scanner controller(s). In case these components are set up properly according to the used hardware environment, it is possible to start marking and motion operations directly from within BeamConstruct:

- Scanner Controller Selection – this combo box has to be used in multihead mode where BeamConstruct controls more than one scanner card at the same time. The selection with this combo box influences the settings of all following parameters. Please note: in multihead and singlehead mode the first scanner controller always has to be set, it is not possible to start e.g. with settings for the second scanner controller and leave the first one empty.
- Scanner Controller – here a scanner card can be chosen out of a list of available scanner controllers. As soon as a scanner card is selected for usage within BeamConstruct, it can be configured by pressing the button “Configure”. Although this configuration is done for the scanner controller card all the parameters like the working area size and position are used to set up BeamConstruct too. For a description of the available scanner controller plug-ins, their usage and parameters, please refer to section “Error: Reference source not foundError: Reference source not found” above
- Initialise on startup – when this option is set, the scanner controller card and – when available and configured – image capture devices, motion-, laser- and process-controllers are initialised on startup

of the software. Normally this is done when the Mark-dialogue is opened for the first time, but when this option is enabled, it happens when BeamConstruct is executed. In general it is not recommended to use this option since it causes possibly unnecessary accesses to the hardware. Only in very rare cases where some external equipment has to be initialised as early as possible (e.g. when some hardware needs to be heated up), this function should be used

- Home position – when this option is set, the scanner is moved back to the XY-coordinates given right beside this checkbox whenever a marking cycle was finished; using this function the head always ends up in a defined position on regular marking cycles, no matter what the last marked vector really was.
- Override manual stop – when a marking operation is stopped manually (e.g. by pressing the stop-button in marking dialogue or by sending a stop via one of the programming interface) all operations are cancelled as fast as possible and a probably used home position is ignored. When this option is enabled, the software tries to reach the home position also in case of a stop so that marking always ends at this defined position.
- Pilot Laser – this section is related to the configuration of a pilot laser that does not perform marking operations but points to the position where the main laser would influence the working piece (as some kind of preview-laser). This pilot laser can be used out of BeamConstruct's marking dialogue directly (please refer below for more details); the related parameters are enabled only in case a scanner controller card was chosen and configured properly
- Output Port – here a port can be chosen that has to be used to turn the pilot laser on and off
- Output Number – here the bit number at the previously chosen output port has to be selected which shall be used to toggle the pilot laser;  
PLEASE NOTE: some scanner controller cards use some of their output bits for specific, reserved purposes, these bits should not be used for the pilot laser to avoid any unwanted and undefined behaviour. As a second precondition it is necessary to connect the pilot lasers on/off input with the output that was selected here.
- Timeout – here a timeout (in unit minutes) can be specified which turns off a pilot laser after this time. Using this function it can be ensured a user does not (accidentally) let a pilot laser run over hours or independently
- Invert Output Logic – inverts the output value for the signal that should turn the pilot laser on, this option needs to be set for some specific pilot lasers to invert their behaviour
- Always on – when this option is set, the pilot laser is kept on all the time, means it is not modulated and not turned off during jumps. Only before next real marking operation it is ensured the pilot laser is turned off since some laser types can't turn on the main laser as long as the pilot is still on.
- Drop home positions – this option can be checked when a project configuration makes use of the home position but the pilot laser should not include this position during preview. When it is set, only the plain geometries are shown during pilot laser preview, when it is set and when a home position is used, both, the geometries and the home position are shown in preview.
- Correction Factor / Correction Offset – pilot lasers normally have a different colour and therefore a different wavelength than the main laser. In case some optics are used within the scan head, this may result in variances in the marked positions due to a different light refraction factor. Using these parameters such variances can be compensated. The correction factor is a factor that modifies the size of the pilot lasers output in X and Y direction while the correction offset modifies its absolute position
- Ready For Marking Signal – here a signal can be defined that is emitted on a digital output of the scanner card as soon as the software is ready for marking. When the additional option “Combine with laser state” is set too, this output no longer signals only the ready-state of the software but combines this with the laser state. So when the laser signals some error or not being ready by some separate digital inputs, the “ready for marking”-signal will not go to high then.
- Output Port – here a port can be chosen that has to be used for the “Ready For Marking” signal
- Output Number – here the bit number at the previously chosen output port has to be selected which shall be used to toggle the “Ready For Marking” signal;

PLEASE NOTE: some scanner controller cards use some of their output bits for specific, reserved purposes, these bits should not be used for this signal in order to avoid any unwanted and undefined behaviour.

- Invert Output Logic – inverts the output value for the ready-signal
- Mark dialogue opened signal – here a signal can be defined that is emitted on a digital output of the scanner card as long as the marking dialogue is open
- Output Port – here a port can be chosen that has to be used for the “mark dialogue open” signal
- Output Number – here the bit number at the previously chosen output port has to be selected which shall be used to toggle the “marking dialogue open” signal;  
PLEASE NOTE: some scanner controller cards use some of their output bits for specific, reserved purposes, these bits should not be used for this signal in order to avoid any unwanted and undefined behaviour.
- Follow-up time – here a time range (in unit seconds) can be specified the “mark dialogue open” signals stays turned on after this dialogue window was closed. When it is reopened again before this time elapses, the output stays active without interruption
- Invert Output Logic – inverts the output value for the active-signal
- Marking Active Signal – here a signal can be defined that is emitted on a digital output of the scanner card as long as marking is in process; this signal is NOT identical to the laser gate which turns laser on and off during such a marking process
- Output Port – here a port can be chosen that has to be used for the “Marking Active” signal
- Output Number – here the bit number at the previously chosen output port has to be selected which shall be used to toggle the “Marking Active” signal;  
PLEASE NOTE: some scanner controller cards use some of their output bits for specific, reserved purposes, these bits should not be used for this signal in order to avoid any unwanted and undefined behaviour.
- Invert Output Logic – inverts the output value for the active-signal
- Interlock Input – this section provides functionality to watch the state of an interlock / emergency stop circuit.  
**PLEASE NOTE: this function is NOT intended to perform any security-related functions, it neither is able to implement emergency stop functionality nor is it able to turn off any hardware in case of an emergency! All these emergency-stop-functions have to be done in hardware only and apart from any software or connected control boards! This function is only able to watch the state of external emergency/interlock circuitry and to ensure reinitialisation of the connected controllers when an external event has occurred, was handled by the external hardware and is no longer active!**  
When this function is used, the input selected with the options below, is expected to be HIGH. In case it is not, no marking operations are possible. When it goes to LOW during a marking operation is running, it is stopped. Once the input was LOW, a possibly opened marking dialogue has to be closed and all connections to used devices are shut-down automatically. Due to the shut-down of the connected devices, as soon as the input is back at HIGH, a complete initialisation of all hardware is performed including referencing operations (if necessary). So this function can ensure all these devices are in a proper and defined state also after an emergency stop.
- Input Port – here an input port can be chosen where the interlock/emergency stop signal can be read at
- Input Number – here the bit number at the previously chosen input port has to be selected which shall be used to watch the emergency stop/interlock signal

When more than one scanner/head was configured, it is possible to do some multihead-related configurations in tab-pane “Multihead”. When only one scanner controller/scanhead exists, these settings do not have any influence on marking operations:

- Preferred head usage – within this matrix it is possible to define which head should be used for what kind of operations. So as long this is really possible for a head and related geometries, the head then will be used for this operation. Here following options can be chosen for a head, search order for suitable heads during marking is from left to right:
  - Automatic (default) – the geometries are assigned to this head, which is suited best to process them. So in case of partial overlapping working fields the automatic operation tries to spread the geometries over the heads in a way so that as long as possible marking operations are done while switching as seldom as possible from one head to an other
  - Contour – the head should try to mark only contours. Only when a contour is located in an area of an other scanhead and outside of the working field of the selected scanhead, or when an other head in “Automatic” mode fits better, this option is ignored and an other scanhead will mark contours
  - Hatch – the head should try to mark only hatches. Only when a hatch is located in an area of an other scanhead and outside of the working field of the selected one, or when an other head in “Automatic” mode fits better, this option is ignored and this other scanhead will mark the related hatch
  - Require trigger – when marking data are available for and have to be sent to this head, a trigger is inserted into the stream of data. So when this option is set, the change from one head to an other is neither done automatic nor do they run in parallel permanently, but an external trigger signal is needed (ExtStart) in order to start output of marking data at the head where this option is set
  - Wait for tool – this option is related to milling output devices which have the ability to send some kind of “Tool change”-command. When this option is set and the software detects a change of the pen during a marking cycle, a) a tool-change command is sent to the related controller and b) a message box pops up for the head where this option is set. This message box asks the user to change the tool and the marking process will not continue until this message box is confirmed

In tab panel Motion Axes it is possible to configure up to two different motion controllers with up to four independent axes each. Both possible motion controllers are identified via names OUT2 and OUT3. Here a valid configuration is required for at least one of them:

- Motion Controller OUT2 or OUT3 – this combo box can be used to select a motion controller that would be accessed whenever a motion control information is assigned to output OUT2 or OUT3. This selection list makes use of several motion plug-ins which are described in detail below. The parameters that are set for such a motion controller are NOT handed over to BeamConstruct fully, so when some of the axes of a motion controller are not enabled here within its settings, they are still be usable and configurable within the related BeamConstruct elements.  
For a description of the available motion controller plug-ins, their usage and parameters, please refer to section “Error: Reference source not foundError: Reference source not found” above
- Bind axis to slices Z-position – this function is related to marking of sliced 3D data. This option should be set for all these axes, which perform a Z-movement which has to have the same height as the slice thickness of a 3D slice group which is processed. Then the height of the slice is set as motion distance automatically – no matter, what value the related element is configured really. So when a 3D model is sliced with different parameters, the motion settings of a globally configured process project (an external project which is executed between slices automatically) do not need to be changed manually. Only the sign of the motion distance of the motion element is kept, means when a negative movement is specified, the negative movement direction will be performed – but using the slice height.  
This option can be set for all axes which in any way are related to a motion operation which depends on the height of a slice.  
Following preconditions have to be met in order to let the motion distance of an axis automatically be set to the current height of a slice:
  - the option “Bind axis to slices Z-position” needs to be set
  - within the related motion element the axis motion needs to be set to relative
  - a positive or negative relative motion distance needs to be set
  - the motion element is part of a sliced 3D group (as child of the related slice) or it is part of a project which is processed between slices (please refer to parameter “Process between slices” below)

- Auto-move back to start position: when this option is set for an axis, its position at the beginning of a marking cycle is stored. As soon as the marking operation has ended, the axis will be moved back to this starting position automatically.
- Key controlled – configure single axes for control via keyboard operation. When this option is enabled, these axes can be moved out of mark-dialogue or “Motion”-panel by pressing ALT-key plus one of the cursor keys up/down/left/right. Then these axes are moved by the step distance and the speed as entered in related input fields.  
Here it is possible to configure up to two axes for this operation mode. When the related checkbox is set for a longitudinal or vertical axis, it can be controlled by the key-combination ALT plus cursor up/down. When the checkbox is set for a horizontal axis, it can be moved step-wise by the key-combination ALT plus cursor up/down in motion panel or mark dialogue.
- Reference axis on start-up – here these axes can be specified that have to be referenced on application start-up or on first opening of devices automatically
- Referencing order – this option is related to the referencing-function which can be enabled with the reference-checkbox as described above. Using these drop-down-lists, an order can be specified in which the axes shall be referenced. The smaller a number is, that is selected for an axis here, the earlier the related axis will be referenced. When different axes make use of the same order number, they are either referenced at the same time or the underlying motion-controller decides which order is suitable.
- Axis alias and orientation – here for the default axes A, B, C and D an own name can be set that is shown e.g. within mark dialogue to identify the axis; additionally for planar axes it can be specified in which direction movement is done, the setting made here influences what kind of arrows are shown for axis movements in mark dialogue
- Don't show warning when referencing axes – when some axes are configured to perform referencing/homing operation on start-up, this may result in an unexpected, surprising and therefore possibly dangerous movement. Because of this prior to referencing operations an information is shown the user has to confirm in order to start the referencing operation. When this option is set, this information is NOT shown, referencing of axes starts immediately and WITHOUT ANY WARNING! Thus this option has to be used carefully and only in case the whole machine is secure and covered completely.
- Process between slices – here a path to an existing project can be given that is used when 3D meshes are sliced; this project can contain a processing sequence that is performed between every slice

The tab-pane “Other hardware” contains the possibility to configure other external devices which may be required for operating a machine which is controlled by BeamConstruct:

- Image capture – this gives the possibility to choose a source for a live background video which is displayed in drawing area behind all vector and bitmap data. This live image can be used to adjust the current marking project to the working piece that is shown by a connected camera.  
To have a background image just as pattern for a laser project, it is recommended not to use such a live image but to import a still image and use it as light-table background (via menu item “Project” → “Import”, please refer to description of import functionality below).  
For a description of the available image and video capture plug-ins, their usage and parameters, please refer to section “Error: Reference source not foundError: Reference source not found” above.
- Position / Size – using these input fields the position and displayed size of the captured video can be specified, it expects the same coordinate system and orientation like it is used for definition of working area.  
Beside of entering values for position and size of the live background image manually, it is also possible to right-click into the drawing area, select option “Modify live background” and then drag and scale the background image like any normal geometric element in order to place and stretch/shrink it as desired
- Display image on manual activation only – when this option is set, the image is not shown all the time, it has to be enabled manually and can be disabled afterwards to continue without a live background image; this can be done via vision menu or via the image button in mark dialogue

- Stop image capture on motion operations – this option may be useful on systems with limited computing power, when it is set, capturing of image data is stopped as long as an external axis is driven
- Stop image capture during marking operations – this option may be useful on systems with limited computing power, when it is set, capturing of image data is stopped as long as marking is in progress
- Scanhead Selection – this combo box is dedicated to multihead configuration too, different to the other combo box which can be found in “Scanner” tab-pane, it is assigned to a scan head. This differentiation is made because one scanner controller / one plug-in defined above may access several scan heads independently, therefore head-specific parameters have to be set separately. The selection made with this box belongs to the following two parameters:
- Additional laser control – here a laser controller plug-in can be selected that communicates with a laser independent from the scanner card. As soon as a laser controller is selected for usage within BeamConstruct, it can be configured by pressing the button “Configure”. This function is useful in case a laser is used that requires control via a different communication channel that does not exist on a scanner controller card. This can be e.g. an Ethernet connection or a serial interface which is used to send control commands to the laser.  
For a description of the available scanner controller plug-ins, their usage and parameters, please refer to section “Error: Reference source not foundError: Reference source not found” above.
- Z-Shifter – here a plug-in can be selected that communicates with a special motion axis to be used for shifting the Z-position, expanding the beam or shifting the lasers focus independent from the scanner card. As soon as such as Z-Shifter is selected for usage within BeamConstruct, it can be configured by pressing the button “Configure”.  
For a description of the available scanner controller plug-ins, their usage and parameters, please refer to section “Error: Reference source not foundError: Reference source not found” above.
- Process Controller – here plug-ins can be selected that are able to control additional process parameters (beside the laser marking related ones). Possible parameters are pressure, temperature, gas flow and others where the current value is read from an external device. Depending on the possibilities and settings of the used plug-ins, other devices can be controlled or switched and/or the currently running marking process can be halted or continued. Also depending on the used plug-ins, the pen settings dialogue may be extended by an additional tab-pane as soon as such a process control plug-in is enabled. This additional tab-pane then contains parameters which are handed over to this plug-in during runtime and dependent on which pen is currently selected.  
As soon as such as Process Control plug-in is selected for usage within BeamConstruct, it can be configured by pressing the button “Configure”. There all parameters can be set according to the capabilities of this plug-in.  
Here several process-control plug-ins can be selected. The combination of these plug-ins allows to access different sensors which are connected via different interfaces. Typically a plug-in supports exactly one type of sensor and in most cases is able to measure one value via this sensor. Now by combining several plug-ins, several types of sensors can be combined and used. So these plug-ins provide some kind of tree-structure:
  - process-control plug ins (several can be chosen and combined freely in this panel)
    - one or more values measured and displayed by this plug in (such as temperature, gas pressure, gas concentration,...); typically one plug-in is responsible for one sensor and one measured value
      - up to four parameters that belong to each measured and displayed value (such as lower threshold, upper threshold, setpoint, ...), which can be configured in settings of the plug in (here their default values can be set) and which can have different values for each pen (the pen-dependent values can be set in pen settings dialogue as soon as such a process-control plug-in was set up)

Tab-pane "Misc" contains options and configuration possibilities that do not belong to any of the previous ones logically:

- Enable IOSelect-Mode – this checkbox can be used to activate the IOSelect functionality (please refer below) and to enable the settings that belong to it:

- Numbered Projects directory – here path to a directory has to be specified that contains the project data to be loaded depending on the current input state
- Card 1 Input Port – the digital input port that has to be used for external selection of next project; this selection is important in case a card provides more than one digital input port; as specified by the name of this parameter, only the first scanner card can be used to read the IOSelect input data from
- Auto-Start Marking – when this option is set, marking of IO-selected project starts automatically after mark-dialogue was opened. Attention: this function may be dangerous and should be handled with care! The start of a marking operation directly after opening the mark-dialogue and without further actions may be unexpected and surprising for the user!
- Auto-optimize vectors on marking – BeamConstruct is able to handle a very high accuracy internally – which is in very most cases much higher than what the controller hardware and scanhead are able to produce on output. When this option is set, the vector data are analysed before they are sent to the scanner controller(s) and all data that are beyond the resolution of the used hardware are removed. Depending on the data to be marked this can save nameable amount of marking time and data.
- Auto-slice 3D-meshes on change – when a 3D-model is imported to be sliced into layers (please refer to section 5.10.2) there are several operations that require reslicing of these data. Depending on the size of the 3D model and the thickness of slices this can be a time-consuming operation and therefore has to be triggered by the user manually. This behaviour can be changed, when this option is set, reslicing of the 3D mesh is done automatically after every operation that requires it. This option should be used only in case the used 3D models are simple enough or the computer hardware is powerful enough, elsewhere it can result in long lasting delays during editing of 3D models and sliced data.
- Show Tab-Panes in Mark-Dialogue – this area consists of a set of checkboxes that correspond to the different tab-panes shown in Mark-Dialogue. Here it is possible to uncheck some of them in order to hide the related tab-panes. This function can be used to not to show unused functions to the end user
- The sub-section “Limits” gives the possibility to configure allowed ranges and behaviour of some pen parameters.  
Here for each scanhead the minimum and maximum value that can be specified within pen settings dialogue can be predefined. The head these limits have to be applied for can be selected via the combobox right above the input fields. The allowed range of following values can be configured for each head separately:
  - Pen Frequency
  - Jump Speed
  - Mark Speed
 Beside of that the behaviour of the related slider can be changed, when check box “Logarithmic” is set, the sliders no longer act linear.  
Option “Allow unrestricted laser on/off delays” disables some logic checks in pen settings: Normally the delays applied to a pen have to follow some logic rules which are automatically checked when the user modifies their values. When this option is set, these logic checks are disabled and illegal laser delay values can be set for a pen. Since this may result in some very strange and undefined behaviour during marking, this function should be used only in some very rare cases where it is clear what happens when such illegal laser parameters are used!
- Parts Counter / Enable – with these two parameters some processing-related limits can be set: this option enables the parts counter; a function which counts how many parts already have been processed and informs the user when a limit was reached; in this case the user has to confirm this limit and needs to reset the counter to continue marking; confirmation and resetting can be done in mark dialogue



- Default part number – here a number of default parts can be specified that is used for the parts counter, it can be overwritten by the user in mark dialogue and requires the parts counter option to be enabled
- Power down laser / Enable – enables a function which gives the possibility to power down a laser when it is unused for a given time
- Power down idle timeout – here a value in seconds can be set which specifies after what idle time a connected laser has to be turned off; this requires the power-down option to be enabled
- Trace log – similar to the Smart Interface (as described in section “5.22 Smart Interface”), this option lets the software write trace information into a log file. This log file can be specified here and makes use of CSV format. Once enabled, all Elements, where the option “Trace data” is enabled, are written into this log file in pairs of columns consisting of the name of the element and the assigned data that have been marked. For serial number elements, these assigned data are the serial number used for a marking operation

Tab-pane “Custom dot” contains configuration possibilities to set up one or more custom dots which then can be used in elements for creating geometries:

- Custom dot – gives the possibility to choose from a list of dot configurations which then can be edited using the parameters below
- Custom dot name – here the selected custom dot can be renamed to give it a unique, easy to identify name
- Custom Dot Style – the following parameters are responsible for defining a style for a customised dot layout that is used by several geometric elements; this drop-down box selects the type of geometry, here polygon (includes a circle) or spiral can be chosen to be used as replacement for simple dots
- Edges/Segments – specifies the number of the edges or segments the custom geometry has to consist of
- Outer Radius – sets the size of the custom geometry
- Inner Radius – this parameter is used only in case of a spiral, it specifies the inner size of it
- Slope – here the same is true, this parameter applies to a spiral only and can be used to set the steepness the spiral grows with
- Draw CCW – here the drawing direction of the custom geometry can be inverted
- Inside To Outside – when this option is set for a spiral, it is drawn starting from its inner point

All parameters from this dialogue are stored within local default settings and within every .BEAMP project file. Thus it may happen a project file is loaded that contains different hardware settings and therefore is not compatible to the local system. To avoid such troubles two functions can be found in menu “Project” that handle the local default data:

- Save as default configuration – saves the current set up as default, whenever the application is started or a new project is generated, this configuration is used automatically; this not only includes the settings done within the project settings but also the number of pens and pen settings
- Load default configuration – loads a saved default configuration, overwrites the current project settings and appends the loaded pen definitions to the currently used ones

### 5.5.1 Scanner Controller Configuration

Dummy scanner controller

This plug-in does not communicate with an external hardware but is a dummy for testing or demonstration purposes. From its configuration possibilities and parameters it is identical to the “HALaser E1803D Scanner Controller Card” but comparing to it, it does not communicate with the E1803D hardware. Instead all its functions return with “OK” or “ready” immediately.

This plug-in is located in flowplugins/libio\_sc\_dummy.

#### G-Code Controller (Ethernet):

**WARNING:** This plug-in is designed to control laser and other equipment which may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured.

This plug-in can be used to access laser-controller or other similar devices that accept G-Code (CNC) data transferred via Ethernet. This is a generic plug-in which does not expect a specific device but provides generic G-Code commands conform to the standard which normally should work with all G-Code-compatible devices. Following parameters can be configured:

- IP and Port – IP and port number of the device to connect with
- Number of Axes – specifies if the scanner controller has to operate in 2D or 3D mode
- Laser type – here the type of used laser has to be set; this parameter corresponds to the option of custom M7xx commands described below

Laser Type	Signals
Generic laser	This is for generic laser types which do not require special control signals and therefore do not make use of any of the M7xx commands; this mode has to be used when the connected controller handles all laser signals itself
CO <sub>2</sub>	stand-by frequency and length are used for initialisation (tickle-pulse after opening the device)  default power control and laser on/off via PWM control (longer pulses cause laser emission, short tickle pulses keep laser in stand-by)  additional definitions in "CO <sub>2</sub> "-tab-pane
YAG Mode 1/2/3	stand-by frequency and length are used for initialisation (tickle-pulse after opening the device)  default power control and laser on/off via PWM control (longer pulses cause laser emission, short tickle pulses keep laser in stand-by)  first pulse killer supported at laser on (mode description see manual of E1701C)
Laser Mode 4	no predefined power control definition of continuously running frequency in "Laser Mode 4"-tab-pane continuously running frequency can be changed by parameter at beginning of mark
SPI (HS)	stand-by frequency is used for initialisation and running independent from jump/mark (continuously running frequency)  frequency can be changed by parameter at beginning of mark  default power control via A0 analogue output  simmer control requires laser controller plug-in and serial interface communication  additional definitions in "SPI"-tab-pane
MOPA (MO, Latch, LP8)	to be used for digitally controlled MOPA laser types which require main oscillator and power control via latched LP8 port  stand-by frequency is used for initialisation and running independent from jump/mark (continuously running frequency with 50% pulse-width at LaserA output)  frequency can be changed by parameter at beginning of mark  default power control via latched LP8 digital outputs

Laser Type	Signals
	additional definitions in "MOPA" tab-pane
Generic mill	<p>To be used when no laser but a mechanical tool like a mill is used</p> <p>no setting of power, frequencies or laser delays necessary</p> <p>maximum speed for Z axis limits speed of tool when diving into the material</p> <p>maximum increment for Z axis (in mm) can be set for automatic calculation of iterations that are necessary for milling into material of a specified thickness</p>

- Number of axes – specifies if 2D or 3D data have to be submitted to the device
- Working area – defines the upper left coordinate of the field that has to be used by the scanner and the total size of the field that is used by the scanner, this parameter together with the left and upper position specifies the coordinates input vectors have to be located within, in case there are input data which define a position outside this field they will not be marked
- Custom initialisation commands – here it is possible to add own, device-specific G-Code commands that are sent during initialisation of the device
- Job start commands – here it is possible to add own, device-specific G-Code commands that are sent to the device at the beginning of each marking cycle
- Tool change commands – here it is possible to add own, device-specific G-Code commands that are sent to the device whenever the pen is changed; this command is typically used in milling mode and can be used to invoke a tool-change at the machine in order to reflect the different pen
- Laser on commands – commands configured here are sent to the device each time the laser has to be turned on; this function can be used to configure own, custom sequences to switch on the connected output device which can be – dependent on the chosen mode – a mill or a laser
- Laser off commands – commands configured here are sent to the device each time the laser has to be turned off; this function can be used to configure own, custom sequences to switch off the connected output device which can be – dependent on the chosen mode – a mill or a laser
- Jump mode – here a mode can be chosen to perform fast jumps to a position while the laser or tool is turned off. It can be chosen between G-Code command "G00" which performs an as fast as possible movement, and "Jump speed" which performs a normal, linear movement using the jump speed specified below
- Include custom M7xx laser commands – when this option is set, some laser-specific commands are included into the generated G-Code which contain parameters and settings which are required for and specific to laser marking operations. Following commands are supported:  
M700 – the laser type, here a decimal number has to be set which specifies the laser type in detail

Laser type	M700 decimal parameter	Corresponding hexadecimal value
CO2	1207959553	0x48000001
YAG1 (Q-Switch and FPK at the beginning)	3355443202	0xC8000002
YAG2 (Q-Switch after FPK at the beginning)	2818572291	0xA8000003
YAG3 (Q-Switch after variable Q-Time)	2818572292	0xA8000004
CRF (Continuously running frequency)	1073741829	0x40000005
MOPA	1207959560	0x48000008

M701 – laser frequency in unit Hz (generic laser types only)

M702 – laser power in unit %\*1000 (generic and MOPA lasers only)

M703 – laser on delay (A) and laser off delay (B) in unit µsec

- M704 – jump speed (A) and mark speed (B) unit mm/min
  - M705 – jump delay (A), mark delay (B) and in-polygon delay (C) in unit  $\mu\text{sec}$
  - M706 – reserved for future use
  - M707 – working area position X and Y in unit specified by G20/G21 command
  - M708 – jump delay (A), mark delay (B) and variable in-polygon delay (C) in unit  $\mu\text{sec}$  (maximum delay at 180 degrees angle)
  - M709 – working area size in X, Y and Z direction in unit specified by G20/G21 command
  - M710 – pulselength in unit nsec (MOPA and CRF lasers with variable pulse width only)
  - M711 – wobble amplitude in X-direction (X) in unit  $\mu\text{m}$ , wobble amplitude in Y-direction (Y) in unit  $\mu\text{m}$ , wobble frequency (C) in unit mHz ( $\text{Hz} \times 1000$ )
  - M712 – reserved for future use
  - M713 – FPK (first pulse) in unit nsec (YAG lasers only)
  - M714 – simmer current in unit  $\% \times 1000$  (SPI laser only)
  - M715 – stand-by frequency (A) in unit Hz and stand-by pulsewidth (B) in unit nsec ( $\text{CO}_2$  and YAG lasers only)
  - M716 – reserved for future use
  - M717 – waveform number (SPI laser only)
  - M718 – laser frequency in unit Hz (A) and laser pulsewidth in unit  $\mu\text{sec}$  (B, needs to fit to period of current laser frequency)
  - M719 – switch main oscillator (MOPA lasers only), 1 for on, 0 for off
  - M720 .. M799 – reserved for future use
- Swap X and Y – swaps the input X and Y coordinate to exchange X and Y axis values of the scanner
  - Mirror X / Mirror Y – mirror the input coordinates in relation to output

The panel “Milling” offers following options:

- Fixed height for jumps – when the plug-in is used with a mill or any other mechanical tool (instead of a laser head), additional movements are necessary that pull out the tool from the material after a mark operation and that move the tool into the material when starting a new mark operation (independent from what the nominal Z-height of the input data is). When this option is set, these additional movements are added automatically and all jumps are done at a fixed height, which can be set with the following parameter:
- Z-position for jumps – here a value (in unit mm) can be added that specifies a secure Z-height for the tool to move at when not marking (milling). Here a value should be entered which is big enough to have a height which is above the working piece

#### G-Code Controller (Serial):

**WARNING:** This plug-in is designed to control laser and other equipment which may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured.

This plug-in can be used to access laser-controller or other similar devices that accept G-Code (CNC) data transferred via serial interface. This plug-in is identical to the previous one (G-Code Controller (Ethernet)) except the following functionality and features:

- serial communication – instead of an IP and port number a serial port is used for communication, thus in general settings tab-pane instead of IP and port number the name and parameters of the serial port have to be configured
- Use GRBL syntax – when this option is set, the plug-in expects to receive responses from the connected controller/sends control commands to the controller in GRBL format
- Position tolerance – when operating in GRBL-mode, two conditions have to be met in order to assume a motion operation is at its end: GRBL returns “Idle” and the returned XY(Z) position is equal to the last requested position. Depending on the resolution and stepwidth of the connected motor it may be possible GRBL is not able to exactly reach the intended target position because it would be located between two steps. For such cases a position tolerance has to be defined. The value given here in +- mm specifies how much the actual position returned from GRBL is allowed to differ from the last given, nominal position while still being a valid target position.

This plug-in is located in flowplugins/libio\_sc\_gcode\_serial.

#### HALaser E1701A Scanner Controller:

**WARNING:** This plug-in is designed to control laser equipment which may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured. Beside of that some laser equipment can be damaged in case it is controlled with wrong signals. Thus it is highly recommended to check the output generated by this plug in using e.g. an oscilloscope to avoid problems caused by wrong configurations. This should be done prior to putting a system into operation for the first time and prior to putting a system into operation after any software update.

This plug-in can be used to access the HALaser Systems E1701A scanner controller cards for controlling a scan head and a connected laser in real-time. Within the configuration dialogue the plug-in and the scanner controller card have to be configured according to the connected devices.

#### Basic configuration parameters:

- IP or Serial Interface - here it has to be configured how and where the controller is accessible. In case of Ethernet connection the IP (in style aaa.bbb.ccc.ddd) has to be entered here, in case of USB-connection the name of the serial interface has to be given ("COMx" for Windows, "/dev/ttyACMx" for Linux where "x" is the number of the serial interface the controller was connected with)
- Ethernet Password – this parameter has to be given only in case Ethernet connection is used; here a password can be specified in order to identify connection to a controller. The same password has to be set within the related E1701 scanner controller. When password specified here and password configured at connected controller do not fit to each other, connection is closed and plug-in is not able to use this controller.
- Analogue voltage range – the E1701A is used to control analogue scanheads. Using this parameter the voltage range this scanhead works with can be selected in range 2,5..10 V (equal to +-2,5 V +-10 V control voltage at the cards outputs)
- Laser type – the type of the connected laser, the field that was selected here directly corresponds to the laser tab panes where some additional, laser-specific settings can be done; for a detailed description of the available options please refer to the manual of E1701 controller and to the manuals of the laser device vendor; within the plug-in the following laser types are available which make use of different parameters:

Laser Type	Signals
CO <sub>2</sub>	<ul style="list-style-type: none"><li>• stand-by frequency and length are used for initialisation (tickle-pulse after opening the device)</li><li>• default power control and laser on/off via PWM control (longer pulses cause laser emission, short tickle pulses keep laser in stand-by)</li><li>• additional definitions in "CO<sub>2</sub>"-tab-pane</li></ul>
YAG Mode 1/2/3	<ul style="list-style-type: none"><li>• stand-by frequency and length are used for initialisation (tickle-pulse after opening the device)</li><li>• default power control and laser on/off via PWM control (longer pulses cause laser emission, short tickle pulses keep laser in stand-by)</li><li>• first pulse killer supported at laser on (mode description see manual of E1701)</li></ul>
Laser Mode 4	<ul style="list-style-type: none"><li>• no predefined power control</li><li>• definition of continuously running frequency in "Laser Mode 4"-tab-pane</li><li>• continuously running frequency can be changed by parameter at beginning of mark</li></ul>
SPI (HS)	<ul style="list-style-type: none"><li>• stand-by frequency is used for initialisation and running independent from</li></ul>

Laser Type	Signals
	<p>jump/mark (continuously running frequency)</p> <ul style="list-style-type: none"> <li>frequency can be changed by parameter at beginning of mark</li> <li>default power control via A0 analogue output</li> <li>simmer control requires laser controller plug-in and serial interface communication</li> <li>additional definitions in "SPI"-tab-pane</li> </ul>
MOPA (MO, Latch, LP8)	<ul style="list-style-type: none"> <li>stand-by frequency is used for initialisation and running independent from jump/mark (continuously running frequency with 50% pulse-width at LaserA output)</li> <li>frequency can be changed by parameter at beginning of mark</li> <li>default power control via latched LP8 digital outputs</li> <li>additional definitions in "MOPA"-tab-pane</li> </ul>

- Additional power output – the laser output power can be controlled via an additional output; the kind of output to be used can be defined here; this is an additional value which does not override or replace the default of the selected laser type. PLEASE NOTE: the possible outputs listed here require specific variants of E1701 scanner card and/or specific extension boards which are optional. Please refer to E1701 scanner controller manual for details which outputs are available together with base- and extension boards.
- Bitmap marking mode – using this combobox the marking mode for scanner bitmaps can be chosen; "Fast" uses the jump speed as long as the laser is turned off within a bitmap and moves with mark speed continuously elsewhere, "Jump'n'shoot" jumps to a pixel position, then shoots with the laser and jumps to the next, "Laser gate modulated" does not make use of any power outputs but turns on/off the laser gate (black and white marking only)
- A0 Correction Factor: this parameter applies to analogue output A0 of LP8 Extensionboard only. It can be used for analogue output voltages in range 0..5V but requires an external power supply which provides 5V exactly (voltages provided via USB are often below of 5V). This factor can be used to adjust the output voltage by giving a correction factor. It of course can not pull the A0 output up to %v when power supply delivers less than 5V but it can be used to adjust voltages below this limit.
- Correction file – a correction file to be used for operation (provided by the vendor of the scan head). Here several formats are supported by the card directly and can be used without any conversion.
- Stand By Frequency – frequency the laser has to be driven with in standby mode (depends on selected laser type); if applicable this frequency is emitted after initialisation of the plug-in
- Stand By Length – pulse length the laser has to be driven with in standby mode (depends on selected laser type); if applicable this pulse width is used after initialisation of the plug-in
- Field Left Position / Field Upper Position – defines the upper left coordinate of the field that has to be used by the scanner
- Field Size – the total size of the field that is used by the scanner, this parameter together with the left and upper position specifies the coordinates input vectors have to be located within, in case there are input data which define a position outside this field they will not be marked
- Swap X and Y – swaps the input X and Y coordinate to exchange X and Y axis values of the scanner
- Mirror X / Mirror Y – mirror the input coordinates in relation to output
- Gain X / Gain Y – linear correction factor in X and Y direction to stretch or shrink the output
- Rotation – rotates the output by the given angle (in unit degree) using the centre of the working area as rotation point
- Slant X / Slant Y – trapezoidal correction angle along X and Y axis

- Use A3 as Digi Out – when this option is set, the analogue output A3 is operated as it would be a digital output

#### 3D:

- Use third axis – this checkbox enables 3D marking mode
- Field depth – the Z-size of the field in unit mm; this value is used for displaying the correct working area in BeamConstruct and for calculation of the additional Z correction
- Z distance – the distance between last mirror of the scanhead and middle position of the working area in unit mm; this value is used for calculation of the additional Z correction
- Mirror Z – mirror the Z-axis input coordinates

#### Signals:

- Mark-In-Progress output – specifies a digital output to be used as "Mark in progress" signal. When this option is set, the related output is set to HIGH as long as a marking operation is running. This option can be used only together with Digi I/O Extension board.
- Wait-External-Trigger output – specifies a digital output to be used as "Wait for external trigger" signal. When this option is set, the related output is set to HIGH as long as the controller has halted operation because it is waiting for an ExtStart signal.
- Laser Ready Input – specifies a digital input to be used as "Laser Ready" signal. When this option is set, marking is possible only in case a HIGH signal is detected at the input specified here. This option can be used only together with Digi I/O Extension board.
- Invert LaserGate – inverts the output logic of the LaserGate output
- Invert LaserA – inverts the output logic of the LaserA output
- Invert LaserB – inverts the output logic of the LaserB output
- Invert LP8 – inverts the output logic of the LP8 parallel output
- Invert MO – inverts the output logic of the MO (main oscillator) output
- Turn off power during jumps – this parameter is relevant for lasers with an unusual high leakage that can't be turned off by LaserGate completely. This typically results in thin marking lines for all movements where the laser should have been turned off. When this option is set, the current power is set to 0 before a jump is performed in order to minimize that leakage during scanner movement with LaserGate at LOW (means no emission). This ensures proper marking result also with such faultily lasers, but extends the marking time (up to 10 usec for every jump)
- Marking on-the-fly – can be used to enable marking on-the-fly operations, here the desired mode can be chosen; "1D" makes use of one encoder input (two signal lines) and compensates in one direction, here the direction and strength is specified by one of the following parameters, "2D" makes use of two encoder inputs (four signal lines) and compensates in both directions X and Y, here the strength is specified by the two following parameters, "Simulated" is intended to be used for testing purposes, here MOTF-functionality is available without an external encoder signal, it is generated by the controller card internally
- On-the-fly factor X / On-the-fly factor Y – specifies the encoder factor to be used along the related axis, the value has to be given in unit  $\mu\text{m}/\text{inc}$

#### MOPA:

This tab-pane contains different settings that are specific to MOPA-lasers and apply to the configuration only when a MOPA-laser was selected. Here following options can be set:

- Alarm input (12) – alarm input at the digital interface to be used for alarm-signal 12 of the laser. When this value is set to a value different than "unused", the alarm state of the related pin at the laser is watched. As long as this pin does not signal "ready", marking is impossible and an error message is shown in the mark dialogue. It is recommended to configure DIn4 here.
- Alarm input (11) – alarm input at the digital interface to be used for alarm-signal 11 of the laser. When this value is set to a value different than "unused", the alarm state of the related pin at the laser is watched. As long as this pin does not signal "ready", marking is impossible and an error message is shown in the mark dialogue. It is recommended to configure DIn5 here.

- Alarm input (16) – alarm input at the digital interface to be used for alarm-signal 16 of the laser. When this value is set to a value different than “unused”, the alarm state of the related pin at the laser is watched. As long as this pin does not signal “ready”, marking is impossible and an error message is shown in the mark dialogue. It is recommended to configure DIn6 here.
- Alarm input (21) – alarm input at the digital interface to be used for alarm-signal 21 of the laser. When this value is set to a value different than “unused”, the alarm state of the related pin at the laser is watched. As long as this pin does not signal “ready”, marking is impossible and an error message is shown in the mark dialogue. It is recommended to configure DIn7 here.
- Turn off MO during jumps – this parameter is relevant for lasers with an unusual high leakage that can't be turned off by LaserGate completely. When this option is set, the main oscillator is turned off before a jump is performed in order to minimize that leakage during scanner movement with no emission
- Set frequency before MO – when this parameter is set, the frequency at LaserA is set before the main oscillator is turned on. This may be necessary for some specific MOPA-lasers in order to let them synchronise to the modulation frequency properly. When this option is set, the following parameter can be configured too:
- Delay before MO – here a value in unit msec can be specified which adds a delay after a new frequency has been set and before the main oscillator is turned on again. This may be necessary for some specific MOPA-lasers that react slow on change of the modulation frequency
- Delay after MO – when a value in unit msec is given here, a delay is added every time after the main oscillator was turned on. The main oscillator typically needs some settling time until the laser can accept further hardware signals. The correct delay time to be added here depends on the laser type. The default value of 10 msec should work with all major laser types.
- Delay before MO off – when a value in unit msec is given here, a delay is added every time before the main oscillator is turned off. This is a value specific to the used type of laser. The majority of laser models shall not need this delay, so this value typically can be set to 0.
- PWM control – typically a MOPA laser runs with a fixed duty cycle of 50%. But for some specific types of lasers it is possible to adjust the pulse-width too. Here a proper mode can be selected which defines what method of setting the pulse width has to be used. The value to be select here depends on the used laser type and its capabilities, so the preferred mode has to be chosen dependent on the used laser. As soon as this option is no longer set to “unused”, the pulse-width-parameter or the APD-index-parameter in pen settings becomes active and can be adjusted for every pen. Following methods of setting the pulse-width are supported:

PWM Type	Remarks
PWM value via LaserB, LP8_1, LP8_2	When this option is set, the pulse-width is transferred to the laser using a serial control information. This serial data transmission is done via the specified laser interface pins. In this mode the pulse-width is generated by the laser.
APD-index via DOut2, DOut3, DOut3	When this option is selected, an APD-index value that specifies a pulse-width is transferred to the laser using a serial control information. This serial data transmission is done via the specified digital interface pins. These pins have to be connected to the laser in this mode and are not allowed to be used for anything else. In this mode the pulse-width is generated by the laser.
PWM signal via LaserA	The pulse-width of the LaserA-signal is changed according to the current pen-settings. In this mode the scanner controller influences the pulse-width directly, it is not generated by the laser.



All following tab-panes contain laser-specific definitions that are used depending on the selected laser type. For information about these parameters and their usage please refer to the manuals and specifications of the used lasers.

This plug-in is located in flowplugins/libio\_e1701a.

#### HALaser E1701C CNC Controller:

**WARNING:** This plug-in is designed to control laser equipment and machinery which may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured. Beside of that some laser equipment can be damaged in case it is controlled with wrong signals. Thus it is highly recommended to check the output generated by this plug in using e.g. an oscilloscope to avoid problems caused by wrong configurations. This should be done prior to putting a system into operation for the first time and prior to putting a system into operation after any software update.

This plug-in can be used to access up to three axes of the HALaser Systems E1701C five-axis CNC controller cards for controlling a XY(Z)-table/plotter/gantry and a connected laser in real-time. Within the configuration dialogue the plug-in and the scanner controller card have to be configured according to the connected devices.

#### Basic configuration parameters:

- IP or Serial Interface - here it has to be configured how and where the controller is accessible. In case of Ethernet connection the IP (in style aaa.bbb.ccc.ddd) has to be entered here, in case of USB-connection the name of the serial interface has to be given ("COMx" for Windows, "/dev/ttyACMx" for Linux where "x" is the number of the serial interface the controller was connected with)
- Ethernet Password - this parameter has to be given only in case Ethernet connection is used; here a password can be specified in order to identify connection to a controller. The same password has to be set within the related E1701C scanner controller. When password specified here and password configured at connected controller do not fit to each other, connection is closed and plug-in is not able to use this controller.
- Laser type – the type of the connected laser, the field that was selected here directly corresponds to the laser tab panes where some additional, laser-specific settings can be done; for a detailed description of the available options please refer to the manual of E1701C controller and to the manuals of the laser device vendor; within the plug-in the following laser types are available which make use of different parameters:

Laser Type	Signals
CO <sub>2</sub>	<ul style="list-style-type: none"> <li>• stand-by frequency and length are used for initialisation (tickle-pulse after opening the device)</li> <li>• default power control and laser on/off via PWM control (longer pulses cause laser emission, short tickle pulses keep laser in stand-by)</li> <li>• additional definitions in "CO<sub>2</sub>"-tab-pane</li> </ul>
YAG Mode 1/2/3	<ul style="list-style-type: none"> <li>• stand-by frequency and length are used for initialisation (tickle-pulse after opening the device)</li> <li>• default power control and laser on/off via PWM control (longer pulses cause laser emission, short tickle pulses keep laser in stand-by)</li> <li>• first pulse killer supported at laser on (mode description see manual of E1701C)</li> </ul>
Laser Mode 4	<ul style="list-style-type: none"> <li>• no predefined power control</li> <li>• definition of continuously running frequency in "Laser Mode 4"-tab-pane</li> <li>• continuously running frequency can be changed by parameter at beginning of</li> </ul>

Laser Type	Signals
	mark
SPI (HS)	<ul style="list-style-type: none"> <li>stand-by frequency is used for initialisation and running independent from jump/mark (continuously running frequency)</li> <li>frequency can be changed by parameter at beginning of mark</li> <li>default power control via A0 analogue output</li> <li>simmer control requires laser controller plug-in and serial interface communication</li> <li>additional definitions in "SPI"-tab-pane</li> </ul>
MOPA (MO, Latch, LP8)	<ul style="list-style-type: none"> <li>to be used for digitally controlled MOPA laser types which require main oscillator and power control via latched LP8 port</li> <li>stand-by frequency is used for initialisation and running independent from jump/mark (continuously running frequency with 50% pulse-width at LaserA output)</li> <li>frequency can be changed by parameter at beginning of mark</li> <li>default power control via latched LP8 digital outputs</li> <li>additional definitions in "MOPA" tab-pane</li> </ul>
Generic mill	<ul style="list-style-type: none"> <li>To be used when no laser but a mechanical tool like a mill is used</li> <li>no setting of power, frequencies or laser delays necessary</li> <li>maximum speed for Z axis limits speed of tool when diving into the material</li> <li>maximum increment for Z axis (in mm) can be set for automatic calculation of iterations that are necessary for milling into material of a specified thickness</li> </ul>

- Additional power ramping – the E1701C controller baseboard comes with a PWM output which provides a power ramping signal at a fixed, predefined frequency. This signal changes its pulse-pause-ratio dependent on the laser power which is necessary for the current movement speed and changes during acceleration and deceleration. When this signal can't be used for some reason, it is possible to define a separate method for power ramping with this checkbox. PLEASE NOTE: the possible outputs listed here require specific variants of E1701C scanner card and/or specific extension boards which are optional. Please refer to E1701C scanner controller manual for details which outputs are available together with base- and extension boards.
- Additional power output – the laser output power can be controlled via an additional output; the kind of output to be used can be defined here; this is an additional value which does not override or replace the default of the selected laser type. PLEASE NOTE: the possible outputs listed here require specific variants of E1701C scanner card and/or specific extension boards which are optional. Please refer to E1701C scanner controller manual for details which outputs are available together with base- and extension boards.
- A0 Correction Factor: this parameter applies to analogue output A0 of LP8 Extensionboard only. It can be used for analogue output voltages in range 0..5V but requires an external power supply which provides 5V exactly (voltages provided via USB are often below of 5V). This factor can be used to adjust the output voltage by giving a correction factor. It of course can not pull the A0 output up to 5V when power supply delivers less than 5V but it can be used to adjust voltages below this limit.
- Stand By Frequency – frequency the laser has to be driven with in standby mode (depends on selected laser type); if applicable this frequency is emitted after initialisation of the plug-in; this signal is available at LaserA output only which requires the LP8 extension board
- Stand By Length – pulse length the laser has to be driven with in standby mode (depends on selected laser type); if applicable this pulse width is used after initialisation of the plug-in; this signal is available at LaserA output only which requires the LP8 extension board

- Field Left Position / Field Upper Position – defines the upper left coordinate of the field that has to be used by the scanner
- Field Size – the total size of the field that is used by the scanner, this parameter together with the left and upper position specifies the coordinates input vectors have to be located within, in case there are input data which define a position outside this field they will not be marked
- Swap X and Y – swaps the input X and Y coordinate to exchange X and Y axis values of the scanner
- Mirror X / Mirror Y – mirror the input coordinates in relation to output
- Gain X / Gain Y - linear correction factor in X and Y direction to stretch or shrink the output
- Rotation - rotates the output by the given angle (in unit degree) using the centre of the working area as rotation point
- Slant X / Slant Y - trapezoidal correction angle along X and Y axis

#### X-Axis/Y-Axis/Z-Axis:

- Enable axis – this checkbox is available for the Z-axis only and enables 3D marking mode
- Factor – specifies a factor that is used to calculate a metric value in millimetres out of the drive-internal position information that is given in unit "increments", the exact factor depends on your hardware set-up, the used gear and the resolution of your stepper motor
- Low Limit – specifies the lower limit in unit millimetres where the axis is allowed to move to at minimum
- High Limit – specifies the upper limit in unit millimetres where the drive is allowed to move to at maximum
- Acceleration – specifies the maximum acceleration for this axis to be used at acceleration, deceleration and when stopping the axis
- Invert direction – inverts the movement direction of the axis while in referencing mode while leaving the movement direction in coordinate system unchanged
- Reference mode – specifies the mode to be used for referencing (=homing) an axis on initialisation of the controller card out of the software
- Reference timeout – is a time-out value in seconds that described after what time referencing has to be stopped when no home switch can be found
- Reference speed (enter) – specifies the speed which has to be used during referencing when entering the reference switch (first part of referencing); this speed should be low enough to securely not to leave the reference switch during deceleration phase again
- Reference speed (leave) – specifies the speed which has to be used during referencing when leaving the previously entered reference switch (second part of referencing); typically this speed should be slower than the enter-speed and is responsible for the accuracy of the referenced position
- Reference position – the position given here is the one which is set to the axis after referencing has been finished successfully, so this value should be equal to the position of the reference switch in the machines coordinate system

#### Milling:

- Fixed height for jumps – when E1701C controller is used with a mill or any other mechanical tool (instead of a laser head), additional movements are necessary that pull out the tool from the material after a mark operation and that move the tool into the material when starting a new mark operation (independent from what the nominal Z-height of the input data is). When this option is set, these additional movements are added automatically and all jumps are done at a fixed height, which can be set with the following parameter:
- Z-position for jumps – here a value (in unit mm) can be added that specifies a secure Z-height for the tool to move at when not marking (milling). Here a value should be entered which is big enough to have a height which is above the working piece

#### Signals:

- Mark-In-Progress output – specifies a digital output to be used as "Mark in progress" signal. When this option is set, the related output is set to HIGH as long as a marking operation is running. This option can be used only together with Digi I/O Extension board.
- Wait-External-Trigger output – specifies a digital output to be used as "Wait for external trigger" signal. When this option is set, the related output is set to HIGH as long as the controller has halted operation because it is waiting for an ExtStart signal.
- Laser Ready Input – specifies a digital input to be used as "Laser Ready" signal. When this option is set, marking is possible only in case a HIGH signal is detected at the input specified here. This option can be used only together with Digi I/O Extension board.
- Invert LaserGate – inverts the output logic of the LaserGate output
- Invert LaserA – inverts the output logic of the LaserA output
- Invert LaserB – inverts the output logic of the LaserB output
- Invert LP8 – inverts the output logic of the LP8 parallel output
- Invert MO – inverts the output logic of the MO (main oscillator) output
- Turn off power during jumps – this parameter is relevant for lasers with an unusual high leakage that can't be turned off by LaserGate completely. This typically results in thin marking lines for all movements where the laser should have been turned off. When this option is set, the current power is set to 0 before a jump is performed in order to minimize that leakage during scanner movement with LaserGate at LOW (means no emission). This ensures proper marking result also with such faultily lasers, but extends the marking time (up to 10 usec for every jump)

#### MOPA:

This tab-pane contains different settings that are specific to MOPA-lasers and apply to the configuration only when a MOPA-laser was selected. Here following options can be set:

- Alarm input (12) – alarm input at the digital interface to be used for alarm-signal 12 of the laser. When this value is set to a value different than "unused", the alarm state of the related pin at the laser is watched. As long as this pin does not signal "ready", marking is impossible and an error message is shown in the mark dialogue. It is recommended to configure DIn4 here.
- Alarm input (11) – alarm input at the digital interface to be used for alarm-signal 11 of the laser. When this value is set to a value different than "unused", the alarm state of the related pin at the laser is watched. As long as this pin does not signal "ready", marking is impossible and an error message is shown in the mark dialogue. It is recommended to configure DIn5 here.
- Alarm input (16) – alarm input at the digital interface to be used for alarm-signal 16 of the laser. When this value is set to a value different than "unused", the alarm state of the related pin at the laser is watched. As long as this pin does not signal "ready", marking is impossible and an error message is shown in the mark dialogue. It is recommended to configure DIn6 here.
- Alarm input (21) – alarm input at the digital interface to be used for alarm-signal 21 of the laser. When this value is set to a value different than "unused", the alarm state of the related pin at the laser is watched. As long as this pin does not signal "ready", marking is impossible and an error message is shown in the mark dialogue. It is recommended to configure DIn7 here.
- Turn off MO during jumps – this parameter is relevant for lasers with an unusual high leakage that can't be turned off by LaserGate completely. When this option is set, the main oscillator is turned off before a jump is performed in order to minimize that leakage during scanner movement with no emission
- Set frequency before MO – when this parameter is set, the frequency at LaserA is set before the main oscillator is turned on. This may be necessary for some specific MOPA-lasers in order to let them synchronise to the modulation frequency properly. When this option is set, the following parameter can be configured too:
- Delay before MO – here a value in unit msec can be specified which adds a delay after a new frequency has been set and before the main oscillator is turned on again. This may be necessary for some specific MOPA-lasers that react slow on change of the modulation frequency

- Delay after MO – when a value in unit msec is given here, a delay is added every time after the main oscillator was turned on. The main oscillator typically needs some settling time until the laser can accept further hardware signals. The correct delay time to be added here depends on the laser type. The default value of 10 msec should work with all major laser types.
- Delay before MO off – when a value in unit msec is given here, a delay is added every time before the main oscillator is turned off. This is a value specific to the used type of laser. The majority of laser models shall not need this delay, so this value typically can be set to 0.
- PWM control – typically a MOPA laser runs with a fixed duty cycle of 50%. But for some specific types of lasers it is possible to adjust the pulse-width too. Here a proper mode can be selected which defines what method of setting the pulse width has to be used. The value to be select here depends on the used laser type and its capabilities, so the preferred mode has to be chosen dependent on the used laser. As soon as this option is no longer set to “unused”, the pulse-width-parameter or the APD-index-parameter in pen settings becomes active and can be adjusted for every pen. Following methods of setting the pulse-width are supported:

PWM Type	Remarks
PWM value via LaserB, LP8_1, LP8_2	When this option is set, the pulse-width is transferred to the laser using a serial control information. This serial data transmission is done via the specified laser interface pins. In this mode the pulse-width is generated by the laser.
APD-index via DOut2, DOut3, DOut3	When this option is selected, an APD-index value that specifies a pulse-width is transferred to the laser using a serial control information. This serial data transmission is done via the specified digital interface pins. These pins have to be connected to the laser in this mode and are not allowed to be used for anything else. In this mode the pulse-width is generated by the laser.
PWM signal via LaserA	The pulse-width of the LaserA-signal is changed according to the current pen-settings. In this mode the scanner controller influences the pulse-width directly, it is not generated by the laser.

All following tab-panes contain laser-specific definitions that are used depending on the selected laser type. For information about these parameters and their usage please refer to the manuals and specifications of the used lasers.

This plug-in is located in flowplugins/libio\_e1701c.

#### HALaser E1701D Scanner Controller:

**WARNING:** This plug-in is designed to control laser equipment which may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured. Beside of that some laser equipment can be damaged in case it is controlled with wrong signals. Thus it is highly recommended to check the output generated by this plug in using e.g. an oscilloscope to avoid problems caused by wrong configurations. This should be done prior to putting a system into operation for the first time and prior to putting a system into operation after any software update.

This plug-in can be used to access the HALaser Systems E1701D scanner controller cards for controlling a scan head and a connected laser in real-time. Within the configuration dialogue the plug-in and the scanner controller card have to be configured according to the connected devices.

Basic configuration parameters:

- IP or Serial Interface - here it has to be configured how and where the controller is accessible. In case of Ethernet connection the IP (in style aaa.bbb.ccc.ddd) has to be entered here, in case of USB-connection the name of the serial interface has to be given ("COMx" for Windows, "/dev/ttyACMx" for Linux where "x" is the number of the serial interface the controller was connected with)
- Ethernet Password - this parameter has to be given only in case Ethernet connection is used; here a password can be specified in order to identify connection to a controller. The same password has to be set within the related E1701 scanner controller. When password specified here and password configured at connected controller do not fit to each other, connection is closed and plug-in is not able to use this controller.
- Laser type – the type of the connected laser, the field that was selected here directly corresponds to the laser tab panes where some additional, laser-specific settings can be done; for a detailed description of the available options please refer to the manual of E1701 controller and to the manuals of the laser device vendor; within the plug-in the following laser types are available which make use of different parameters:

Laser Type	Signals
CO <sub>2</sub>	<ul style="list-style-type: none"> <li>• stand-by frequency and length are used for initialisation (tickle-pulse after opening the device)</li> <li>• default power control and laser on/off via PWM control (longer pulses cause laser emission, short tickle pulses keep laser in stand-by)</li> <li>• additional definitions in "CO<sub>2</sub>"-tab-pane</li> </ul>
YAG Mode 1/2/3	<ul style="list-style-type: none"> <li>• stand-by frequency and length are used for initialisation (tickle-pulse after opening the device)</li> <li>• default power control and laser on/off via PWM control (longer pulses cause laser emission, short tickle pulses keep laser in stand-by)</li> <li>• first pulse killer supported at laser on (mode description see manual of E1701)</li> </ul>
Laser Mode 4	<ul style="list-style-type: none"> <li>• no predefined power control</li> <li>• definition of continuously running frequency in "Laser Mode 4"-tab-pane</li> <li>• continuously running frequency can be changed by parameter at beginning of mark</li> </ul>
SPI (HS)	<ul style="list-style-type: none"> <li>• stand-by frequency is used for initialisation and running independent from jump/mark (continuously running frequency)</li> <li>• frequency can be changed by parameter at beginning of mark</li> <li>• default power control via A0 analogue output</li> <li>• simmer control requires laser controller plug-in and serial interface communication</li> <li>• additional definitions in "SPI"-tab-pane</li> </ul>
MOPA (MO, Latch, LP8)	<ul style="list-style-type: none"> <li>• to be used for digitally controlled MOPA laser types which require main oscillator and power control via latched LP8 port</li> <li>• stand-by frequency is used for initialisation and running independent from jump/mark (continuously running frequency with 50% pulse-width at LaserA output)</li> <li>• frequency can be changed by parameter at beginning of mark</li> <li>• default power control via latched LP8 digital outputs</li> <li>• additional definitions in "MOPA"-tab-pane</li> </ul>

- Additional power output – the laser output power can be controlled via an additional output; the kind of output to be used can be defined here; this is an additional value which does not override or replace the default of the selected laser type. PLEASE NOTE: the possible outputs listed here require specific variants of E1701 scanner card and/or specific extension boards which are optional. Please refer to E1701 scanner controller manual for details which outputs are available together with base- and extension boards.
- Bitmap marking mode – using this combobox the marking mode for scanner bitmaps can be chosen; “Fast” uses the jump speed as long as the laser is turned off within a bitmap and moves with mark speed continuously elsewhere, “Jump’n’shoot” jumps to a pixel position, then shoots with the laser and jumps to the next, “Laser gate modulated” does not make use of any power outputs but turns on/off the laser gate (black and white marking only)
- A0 Correction Factor: this parameter applies to analogue output A0 of LP8 Extensionboard only. It can be used for analogue output voltages in range 0..5V but requires an external power supply which provides 5V exactly (voltages provided via USB are often below of 5V). This factor can be used to adjust the output voltage by giving a correction factor. It of course can not pull the A0 output up to 5V when power supply delivers less than 5V but it can be used to adjust voltages below this limit.
- Correction file – a correction file to be used for operation (provided by the vendor of the scan head). Here several formats are supported by the card directly and can be used without any conversion.
- Stand By Frequency – frequency the laser has to be driven with in standby mode (depends on selected laser type); if applicable this frequency is emitted after initialisation of the plug-in
- Stand By Length – pulse length the laser has to be driven with in standby mode (depends on selected laser type); if applicable this pulse width is used after initialisation of the plug-in
- Field Left Position / Field Upper Position – defines the upper left coordinate of the field that has to be used by the scanner
- Field Size – the total size of the field that is used by the scanner, this parameter together with the left and upper position specifies the coordinates input vectors have to be located within, in case there are input data which define a position outside this field they will not be marked
- Swap X and Y – swaps the input X and Y coordinate to exchange X and Y axis values of the scanner
- Mirror X / Mirror Y – mirror the input coordinates in relation to output
- Gain X / Gain Y – linear correction factor in X and Y direction to stretch or shrink the output
- Rotation – rotates the output by the given angle (in unit degree) using the centre of the working area as rotation point
- Slant X / Slant Y – trapezoidal correction angle along X and Y axis
- Offset X / Offset Y – linear correction to offset given coordinates along X and Y axis

#### 3D:

- Use third axis – this checkbox enables 3D marking mode
- Field depth – the Z-size of the field in unit mm; this value is used for displaying the correct working area in BeamConstruct and for calculation of the additional Z correction
- Z distance – the distance between last mirror of the scanhead and middle position of the working area in unit mm; this value is used for calculation of the additional Z correction
- Mirror Z – mirror the Z-axis input coordinates
- Gain Z – linear correction factor in Z direction to stretch or shrink the output
- Offset Z – linear correction to offset given coordinates along Z axis

#### Signals:

- Mark-In-Progress output – specifies a digital output to be used as "Mark in progress" signal. When this option is set, the related output is set to HIGH as long as a marking operation is running. This option can be used only together with Digi I/O Extension board.

- Wait-External-Trigger output – specifies a digital output to be used as "Wait for external trigger" signal. When this option is set, the related output is set to HIGH as long as the controller has halted operation because it is waiting for an ExtStart signal.
- Laser Ready Input – specifies a digital input to be used as "Laser Ready" signal. When this option is set, marking is possible only in case a HIGH signal is detected at the input specified here. This option can be used only together with Digi I/O Extension board.
- Invert LaserGate – inverts the output logic of the LaserGate output
- Invert LaserA – inverts the output logic of the LaserA output
- Invert LaserB – inverts the output logic of the LaserB output
- Invert LP8 – inverts the output logic of the LP8 parallel output
- Invert MO – inverts the output logic of the MO (main oscillator) output
- Turn off power during jumps – this parameter is relevant for lasers with an unusual high leakage that can't be turned off by LaserGate completely. This typically results in thin marking lines for all movements where the laser should have been turned off. When this option is set, the current power is set to 0 before a jump is performed in order to minimize that leakage during scanner movement with LaserGate at LOW (means no emission). This ensures proper marking result also with such faultily lasers, but extends the marking time (up to 10 usec for every jump)
- Halt with ExtStart – when this option is checked, the ExtStart-input is used to halt the current marking operation. To allow marking, ExtStart needs to be at HIGH. When it goes to LOW, marking is halted the next time the laser is turned of (typically at the next jump). When ExtStart goes back to HIGH, marking is continued at the position where it was halted before except the whole operation was stopped in the meantime. When this mode is enabled, external triggers still can be applied, as they react on a rising edge at ExtStart, the input signal has to go to LOW and back to HIGH in such a case.
- Marking on-the-fly – can be used to enable marking on-the-fly operations, here the desired mode can be chosen; "1D" makes use of one encoder input (two signal lines) and compensates in one direction, here the direction and strength is specified by one of the following parameters, "2D" makes use of two encoder inputs (four signal lines) and compensates in both directions X and Y, here the strength is specified by the two following parameters, "Simulated" is intended to be used for testing purposes, here MOTF-functionality is available without an external encoder signal, it is generated by the controller card internally
- On-the-fly factor X / On-the-fly factor Y – specifies the encoder factor to be used along the related axis, the value has to be given in unit  $\mu\text{m}/\text{inc}$

#### MOPA:

This tab-pane contains different settings that are specific to MOPA-lasers and apply to the configuration only when a MOPA-laser was selected. Here following options can be set:

- Alarm input (12) – alarm input at the digital interface to be used for alarm-signal 12 of the laser. When this value is set to a value different than "unused", the alarm state of the related pin at the laser is watched. As long as this pin does not signal "ready", marking is impossible and an error message is shown in the mark dialogue. It is recommended to configure DIn4 here.
- Alarm input (11) – alarm input at the digital interface to be used for alarm-signal 11 of the laser. When this value is set to a value different than "unused", the alarm state of the related pin at the laser is watched. As long as this pin does not signal "ready", marking is impossible and an error message is shown in the mark dialogue. It is recommended to configure DIn5 here.
- Alarm input (16) – alarm input at the digital interface to be used for alarm-signal 16 of the laser. When this value is set to a value different than "unused", the alarm state of the related pin at the laser is watched. As long as this pin does not signal "ready", marking is impossible and an error message is shown in the mark dialogue. It is recommended to configure DIn6 here.
- Alarm input (21) – alarm input at the digital interface to be used for alarm-signal 21 of the laser. When this value is set to a value different than "unused", the alarm state of the related pin at the laser is watched. As long as this pin does not signal "ready", marking is impossible and an error message is shown in the mark dialogue. It is recommended to configure DIn7 here.



- Turn off MO during jumps – this parameter is relevant for lasers with an unusual high leakage that can't be turned off by LaserGate completely. When this option is set, the main oscillator is turned off before a jump is performed in order to minimize that leakage during scanner movement with no emission
- Set frequency before MO – when this parameter is set, the frequency at LaserA is set before the main oscillator is turned on. This may be necessary for some specific MOPA-lasers in order to let them synchronise to the modulation frequency properly. When this option is set, the following parameter can be configured too:
- Delay before MO – here a value in unit msec can be specified which adds a delay after a new frequency has been set and before the main oscillator is turned on again. This may be necessary for some specific MOPA-lasers that react slow on change of the modulation frequency
- Delay after MO – when a value in unit msec is given here, a delay is added every time after the main oscillator was turned on. The main oscillator typically needs some settling time until the laser can accept further hardware signals. The correct delay time to be added here depends on the laser type. The default value of 10 msec should work with all major laser types.
- Delay before MO off – when a value in unit msec is given here, a delay is added every time before the main oscillator is turned off. This is a value specific to the used type of laser. The majority of laser models shall not need this delay, so this value typically can be set to 0.
- PWM control – typically a MOPA laser runs with a fixed duty cycle of 50%. But for some specific types of lasers it is possible to adjust the pulse-width too. Here a proper mode can be selected which defines what method of setting the pulse width has to be used. The value to be select here depends on the used laser type and its capabilities, so the preferred mode has to be chosen dependent on the used laser. As soon as this option is no longer set to “unused”, the pulse-width-parameter or the APD-index-parameter in pen settings becomes active and can be adjusted for every pen. Following methods of setting the pulse-width are supported:

PWM Type	Remarks
PWM value via LaserB, LP8_1, LP8_2	When this option is set, the pulse-width is transferred to the laser using a serial control information. This serial data transmission is done via the specified laser interface pins. In this mode the pulse-width is generated by the laser.
APD-index via DOut2, DOut3, DOut3	When this option is selected, an APD-index value that specifies a pulse-width is transferred to the laser using a serial control information. This serial data transmission is done via the specified digital interface pins. These pins have to be connected to the laser in this mode and are not allowed to be used for anything else. In this mode the pulse-width is generated by the laser.
PWM signal via LaserA	The pulse-width of the LaserA-signal is changed according to the current pen-settings. In this mode the scanner controller influences the pulse-width directly, it is not generated by the laser.

All following tab-panes contain laser-specific definitions that are used depending on the selected laser type. For information about these parameters and their usage please refer to the manuals and specifications of the used lasers.

This plug-in is located in flowplugins/libio\_e1701.

HALaser E1702S Scanner Controller:

**WARNING:** This plug-in is designed to control laser equipment which may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured. Beside of that some laser equipment can be damaged in case it is controlled with wrong signals. Thus it is highly recommended to check the output generated by this plug in using e.g. an oscilloscope to avoid problems caused by wrong configurations. This should be done prior to putting a system into operation for the first time and prior to putting a system into operation after any software update.

This plug-in can be used to access the HALaser Systems E1702D scanner controller cards for controlling a scan head and a connected laser in real-time. Within the configuration dialogue the plug-in and the scanner controller card have to be configured according to the connected devices.

Basic configuration parameters:

- **IP or Serial Interface** - here it has to be configured how and where the controller is accessible. In case of Ethernet connection the IP (in style aaa.bbb.ccc.ddd) has to be entered here, in case of USB-connection the name of the serial interface has to be given ("COMx" for Windows, "/dev/ttyACMx" for Linux where "x" is the number of the serial interface the controller was connected with)
- **Ethernet Password** - this parameter has to be given only in case Ethernet connection is used; here a password can be specified in order to identify connection to a controller. The same password has to be set within the related E1702 scanner controller. When password specified here and password configured at connected controller do not fit to each other, connection is closed and plug-in is not able to use this controller.
- **Scanhead mode** – here the operation mode of the interface to the scanhead can be chosen. The value selected here needs to fit to the scanhead and to the wiring between controller card and scanhead
- **Laser type** – the type of the connected laser, the field that was selected here directly corresponds to the laser tab panes where some additional, laser-specific settings can be done; for a detailed description of the available options please refer to the manual of E1702 controller and to the manuals of the laser device vendor; within the plug-in the following laser types are available which make use of different parameters:

Laser Type	Signals
CO <sub>2</sub>	<ul style="list-style-type: none"> <li>• stand-by frequency and length are used for initialisation (tickle-pulse after opening the device)</li> <li>• default power control and laser on/off via PWM control (longer pulses cause laser emission, short tickle pulses keep laser in stand-by)</li> <li>• additional definitions in "CO<sub>2</sub>"-tab-pane</li> </ul>
YAG Mode 1/2/3	<ul style="list-style-type: none"> <li>• stand-by frequency and length are used for initialisation (tickle-pulse after opening the device)</li> <li>• default power control and laser on/off via PWM control (longer pulses cause laser emission, short tickle pulses keep laser in stand-by)</li> <li>• first pulse killer supported at laser on (mode description see manual of E1702)</li> </ul>
MOPA (MO, Latch, LP8)	<ul style="list-style-type: none"> <li>• to be used for digitally controlled MOPA laser types which require main oscillator and power control via latched LP8 port</li> <li>• stand-by frequency is used for initialisation and running independent from jump/mark (continuously running frequency with 50% pulse-width at LaserA output)</li> <li>• frequency can be changed by parameter at beginning of mark</li> <li>• default power control via latched LP8 digital outputs</li> <li>• additional definitions in "MOPA"-tab-pane</li> </ul>

- Additional power output – the laser output power can be controlled via an additional output; the kind of output to be used can be defined here; this is an additional value which does not override or replace the default of the selected laser type. PLEASE NOTE: the possible outputs listed here require specific variants of E1702 scanner card and/or specific extension boards which are optional. Please refer to E1702 scanner controller manual for details which outputs are available together with base- and extension boards.
- Bitmap marking mode – using this combobox the marking mode for scanner bitmaps can be chosen; “Fast” uses the jump speed as long as the laser is turned off within a bitmap and moves with mark speed continuously elsewhere, “Jump’n’shoot” jumps to a pixel position, then shoots with the laser and jumps to the next, “Laser gate modulated” does not make use of any power outputs but turns on/off the laser gate (black and white marking only)
- Correction file – a correction file to be used for operation (provided by the vendor of the scan head). Here several formats are supported by the card directly and can be used without any conversion.
- Stand By Frequency – frequency the laser has to be driven with in standby mode (depends on selected laser type); if applicable this frequency is emitted after initialisation of the plug-in
- Stand By Length – pulse length the laser has to be driven with in standby mode (depends on selected laser type); if applicable this pulse width is used after initialisation of the plug-in
- Field Left Position / Field Upper Position – defines the upper left coordinate of the field that has to be used by the scanner
- Field Size – the total size of the field that is used by the scanner, this parameter together with the left and upper position specifies the coordinates input vectors have to be located within, in case there are input data which define a position outside this field they will not be marked
- Swap X and Y – swaps the input X and Y coordinate to exchange X and Y axis values of the scanner
- Mirror X / Mirror Y – mirror the input coordinates in relation to output
- Gain X / Gain Y - linear correction factor in X and Y direction to stretch or shrink the output
- Rotation - rotates the output by the given angle (in unit degree) using the centre of the working area as rotation point
- Slant X / Slant Y - trapezoidal correction angle along X and Y axis

#### Signals:

- Mark-In-Progress output – specifies a digital output to be used as "Mark in progress" signal. When this option is set, the related output is set to HIGH as long as a marking operation is running. This option can be used only together with Digi I/O Extension board.
- Wait-External-Trigger output – specifies a digital output to be used as "Wait for external trigger" signal. When this option is set, the related output is set to HIGH as long as the controller has halted operation because it is waiting for an ExtStart signal.
- Laser Ready Input – specifies a digital input to be used as "Laser Ready" signal. When this option is set, marking is possible only in case a HIGH signal is detected at the input specified here. This option can be used only together with Digi I/O Extension board.
- Invert LaserGate – inverts the output logic of the LaserGate output
- Invert LaserA – inverts the output logic of the LaserA output
- Invert LaserB – inverts the output logic of the LaserB output
- Invert LP8 – inverts the output logic of the LP8 parallel output
- Invert MO – inverts the output logic of the MO (main oscillator) output
- Turn off power during jumps – this parameter is relevant for lasers with an unusual high leakage that can't be turned off by LaserGate completely. This typically results in thin marking lines for all movements where the laser should have been turned off. When this option is set, the current power is set to 0 before a jump is performed in order to minimize that leakage during scanner movement with LaserGate at LOW (means no emission). This ensures proper marking result also with such faultily lasers, but extends the marking time (up to 10 usec for every jump)

- Halt with ExtStart – when this option is checked, the ExtStart-input is used to halt the current marking operation. To allow marking, ExtStart needs to be at HIGH. When it goes to LOW, marking is halted the next time the laser is turned of (typically at the next jump). When ExtStart goes back to HIGH, marking is continued at the position where it was halted before except the whole operation was stopped in the meantime. When this mode is enabled, external triggers still can be applied, as they react on a rising edge at ExtStart, the input signal has to go to LOW and back to HIGH in such a case.
- Marking on-the-fly – can be used to enable marking on-the-fly operations, here the desired mode can be chosen; “1D” makes use of one encoder input (two signal lines) and compensates in one direction, here the direction and strength is specified by one of the following parameters, “2D” makes use of two encoder inputs (four signal lines) and compensates in both directions X and Y, here the strength is specified by the two following parameters, “Simulated” is intended to be used for testing purposes, here MOTF-functionality is available without an external encoder signal, it is generated by the controller card internally
- On-the-fly factor X / On-the-fly factor Y – specifies the encoder factor to be used along the related axis, the value has to be given in unit  $\mu\text{m}/\text{inc}$

#### MOPA:

This tab-pane contains different settings that are specific to MOPA-lasers and apply to the configuration only when a MOPA-laser was selected. Here following options can be set:

- Alarm input (12) – alarm input at the digital interface to be used for alarm-signal 12 of the laser. When this value is set to a value different than “unused”, the alarm state of the related pin at the laser is watched. As long as this pin does not signal “ready”, marking is impossible and an error message is shown in the mark dialogue. It is recommended to configure DIn4 here.
- Alarm input (11) – alarm input at the digital interface to be used for alarm-signal 11 of the laser. When this value is set to a value different than “unused”, the alarm state of the related pin at the laser is watched. As long as this pin does not signal “ready”, marking is impossible and an error message is shown in the mark dialogue. It is recommended to configure DIn5 here.
- Alarm input (16) – alarm input at the digital interface to be used for alarm-signal 16 of the laser. When this value is set to a value different than “unused”, the alarm state of the related pin at the laser is watched. As long as this pin does not signal “ready”, marking is impossible and an error message is shown in the mark dialogue. It is recommended to configure DIn6 here.
- Alarm input (21) – alarm input at the digital interface to be used for alarm-signal 21 of the laser. When this value is set to a value different than “unused”, the alarm state of the related pin at the laser is watched. As long as this pin does not signal “ready”, marking is impossible and an error message is shown in the mark dialogue. It is recommended to configure DIn7 here.
- Turn off MO during jumps – this parameter is relevant for lasers with an unusual high leakage that can't be turned off by LaserGate completely. When this option is set, the main oscillator is turned off before a jump is performed in order to minimize that leakage during scanner movement with no emission
- Set frequency before MO – when this parameter is set, the frequency at LaserA is set before the main oscillator is turned on. This may be necessary for some specific MOPA-lasers in order to let them synchronise to the modulation frequency properly. When this option is set, the following parameter can be configured too:
- Delay before MO – here a value in unit msec can be specified which adds a delay after a new frequency has been set and before the main oscillator is turned on again. This may be necessary for some specific MOPA-lasers that react slow on change of the modulation frequency
- Delay after MO – when a value in unit msec is given here, a delay is added every time after the main oscillator was turned on. The main oscillator typically needs some settling time until the laser can accept further hardware signals. The correct delay time to be added here depends on the laser type. The default value of 10 msec should work with all major laser types.
- Delay before MO off – when a value in unit msec is given here, a delay is added every time before the main oscillator is turned off. This is a value specific to the used type of laser. The majority of laser models shall not need this delay, so this value typically can be set to 0.

- PWM control – typically a MOPA laser runs with a fixed duty cycle of 50%. But for some specific types of lasers it is possible to adjust the pulse-width too. Here a proper mode can be selected which defines what method of setting the pulse width has to be used. The value to be select here depends on the used laser type and its capabilities, so the preferred mode has to be chosen dependent on the used laser. As soon as this option is no longer set to “unused”, the pulse-width-parameter or the APD-index-parameter in pen settings becomes active and can be adjusted for every pen. Following methods of setting the pulse-width are supported:

PWM Type	Remarks
PWM value via LaserB, LP8_1, LP8_2	When this option is set, the pulse-width is transferred to the laser using a serial control information. This serial data transmission is done via the specified laser interface pins. In this mode the pulse-width is generated by the laser.
APD-index via DOut2, DOut3, DOut3	When this option is selected, an APD-index value that specifies a pulse-width is transferred to the laser using a serial control information. This serial data transmission is done via the specified digital interface pins. These pins have to be connected to the laser in this mode and are not allowed to be used for anything else. In this mode the pulse-width is generated by the laser.
PWM signal via LaserA	The pulse-width of the LaserA-signal is changed according to the current pen-settings. In this mode the scanner controller influences the pulse-width directly, it is not generated by the laser.

All following tab-panes contain laser-specific definitions that are used depending on the selected laser type. For information about these parameters and their usage please refer to the manuals and specifications of the used lasers.

This plug-in is located in flowplugins/libio\_e1702s.

#### HALaser E1803D Scanner Controller:

**WARNING:** This plug-in is designed to control laser equipment which may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured. Beside of that some laser equipment can be damaged in case it is controlled with wrong signals. Thus it is highly recommended to check the output generated by this plug in using e.g. an oscilloscope to avoid problems caused by wrong configurations. This should be done prior to putting a system into operation for the first time and prior to putting a system into operation after any software update.

This plug-in can be used to access the HALaser Systems E1803D scanner controller cards for controlling a scan head and a connected laser in real-time. Within the configuration dialogue the plug-in and the scanner controller card have to be configured according to the connected devices.

Basic configuration parameters:

- IP or Serial Interface – here it has to be configured how and where the controller is accessible. In case of Ethernet connection the IP (in style aaa.bbb.ccc.ddd) has to be entered here, in case of USB-connection the name of the serial interface has to be given ("COMx" for Windows, "/dev/ttyACMx" for Linux where "x" is the number of the serial interface the controller was connected with)
- Ethernet Password – this parameter has to be given only in case Ethernet connection is used; here a password can be specified in order to identify connection to a controller. The same password has to

be set within the related E1803 scanner controller configuration file. When password specified here and password configured at connected controller do not fit to each other, connection is closed and plug-in is not able to use this controller.

- Laser type – the type of the connected laser, the field that was selected here directly corresponds to the laser tab panes where some additional, laser-specific settings can be done; for a detailed description of the available options please refer to the manual of E1803 controller and to the manuals of the laser device vendor; within the plug-in the following laser types are available which make use of different parameters:

Laser Type	Signals
CO <sub>2</sub>	<ul style="list-style-type: none"> <li>• stand-by frequency and length are used for initialisation (tickle-pulse after opening the device)</li> <li>• default power control and laser on/off via PWM control (longer pulses cause laser emission, short tickle pulses keep laser in stand-by)</li> <li>• additional definitions in “CO<sub>2</sub>”-tab-pane</li> </ul>
YAG Mode 1/2/3	<ul style="list-style-type: none"> <li>• stand-by frequency and length are used for initialisation (tickle-pulse after opening the device)</li> <li>• default power control and laser on/off via PWM control (longer pulses cause laser emission, short tickle pulses keep laser in stand-by)</li> <li>• first pulse killer supported at laser on (mode description see manual of E1803)</li> </ul>
Laser Mode 4	<ul style="list-style-type: none"> <li>• no predefined power control</li> <li>• definition of continuously running frequency in “Laser Mode 4”-tab-pane</li> <li>• continuously running frequency can be changed by parameter at beginning of mark</li> </ul>
SPI (HS)	<ul style="list-style-type: none"> <li>• stand-by frequency is used for initialisation and running independent from jump/mark (continuously running frequency)</li> <li>• frequency can be changed by parameter at beginning of mark</li> <li>• default power control via A0 analogue output</li> <li>• default simmer control via A1 analogue output</li> <li>• additional definitions in “SPI”-tab-pane</li> </ul>
MOPA (MO, Latch, LP8)	<ul style="list-style-type: none"> <li>• to be used for digitally controlled MOPA laser types which require main oscillator and power control via latched LP8 port</li> <li>• stand-by frequency is used for initialisation and running independent from jump/mark (continuously running frequency with 50% pulse-width at LaserA output)</li> <li>• frequency can be changed by parameter at beginning of mark</li> <li>• default power control via latched LP8 digital outputs</li> <li>• additional definitions in “MOPA”-tab-pane</li> </ul>
MOPA (MO, A0)	<ul style="list-style-type: none"> <li>• to be used for analogue controlled MOPA laser types which require main oscillator and power control via analogue interface</li> <li>• default power control via A0 analogue output</li> <li>• RS232 serial interface connection used to read and reset error states</li> <li>• additional definitions in “MOPA”-tab-pane</li> </ul>

- Additional power output – the laser output power can be controlled via an additional output; the kind of output to be used can be defined here; this is an additional value which does not override or replace the default of the selected laser type.
- Bitmap marking mode – using this combobox the marking mode for scanner bitmaps can be chosen; “Fast” uses the jump speed as long as the laser is turned off within a bitmap and moves with mark speed continuously elsewhere, “Jump'n'shoot” jumps to a pixel position, then shoots with the laser and jumps to the next, “Laser gate modulated” does not make use of any power outputs but turns on/off the laser gate (black and white marking only)
- Correction file – a correction file to be used for operation (provided by the vendor of the scan head). Here several formats are supported by the card directly and can be used without any conversion.
- Stand By Frequency – frequency the laser has to be driven with in standby mode (depends on selected laser type); if applicable this frequency is emitted after initialisation of the plug-in
- Stand By Length – pulse length the laser has to be driven with in standby mode (depends on selected laser type); if applicable this pulse width is used after initialisation of the plug-in
- Field Left Position / Field Upper Position – defines the upper left coordinate of the field that has to be used by the scanner
- Field Size – the total size of the field that is used by the scanner, this parameter together with the left and upper position specifies the coordinates input vectors have to be located within, in case there are input data which define a position outside this field they will not be marked
- Swap X and Y – swaps the input X and Y coordinate to exchange X and Y axis values of the scanner
- Mirror X / Mirror Y – mirror the input coordinates in relation to output
- Gain X / Gain Y – linear correction factor in X and Y direction to stretch or shrink the output
- Rotation – rotates the output by the given angle (in unit degree) using the centre of the working area as rotation point
- Slant X / Slant Y – trapezoidal correction angle along X and Y axis
- Offset X / Offset Y – linear correction to offset given coordinates along X and Y axis

#### 3D:

- Use third axis – this checkbox enables 3D marking mode
- Field depth – the Z-size of the field in unit mm; this value is used for displaying the correct working area in BeamConstruct and for calculation of the additional Z correction
- Z distance – the distance between last mirror of the scanhead and middle position of the working area in unit mm; this value is used for calculation of the additional Z correction
- Mirror Z – mirror the Z-axis input coordinates
- Gain Z – linear correction factor in Z direction to stretch or shrink the output
- Offset Z – linear correction to offset given coordinates along Z axis

#### Signals:

- Mark-In-Progress output – specifies a digital output to be used as "Mark in progress" signal. When this option is set, the related output is set to HIGH as long as a marking operation is running.
- Wait-External-Trigger output – specifies a digital output to be used as "Wait for external trigger" signal. When this option is set, the related output is set to HIGH as long as the controller has halted operation because it is waiting for an ExtStart signal.
- Laser Ready Input – specifies a digital input to be used as "Laser Ready" signal. When this option is set, marking is possible only in case a HIGH signal is detected at the input specified here.
- Invert LaserGate – inverts the output logic of the LaserGate output
- Invert LaserA – inverts the output logic of the LaserA output
- Invert LaserB – inverts the output logic of the LaserB output
- Invert LP8 – inverts the output logic of the LP8 parallel output

- Invert MO – inverts the output logic of the MO (main oscillator) output
- Turn off power during jumps – this parameter is relevant for lasers with an unusual high leakage that can't be turned off by LaserGate completely. This typically results in thin marking lines for all movements where the laser should have been turned off. When this option is set, the current power is set to 0 before a jump is performed in order to minimize that leakage during scanner movement with LaserGate at LOW (means no emission). This ensures proper marking result also with such faultily lasers, but extends the marking time (up to 10 usec for every jump)
- Halt with ExtStart – when this option is checked, the ExtStart-input is used to halt the current marking operation. To allow marking, ExtStart needs to be at HIGH. When it goes to LOW, marking is halted the next time the laser is turned of (typically at the next jump). When ExtStart goes back to HIGH, marking is continued at the position where it was halted before except the whole operation was stopped in the meantime. When this mode is enabled, external triggers still can be applied, as they react on a rising edge at ExtStart, the input signal has to go to LOW and back to HIGH in such a case.
- Marking on-the-fly – can be used to enable marking on-the-fly operations, here the desired mode can be chosen; “1D” makes use of one encoder input (two signal lines) and compensates in one direction, here the direction and strength is specified by one of the following parameters, “2D” makes use of two encoder inputs (four signal lines) and compensates in both directions X and Y, here the strength is specified by the two following parameters, “Simulated” is intended to be used for testing purposes, here MOTF-functionality is available without an external encoder signal, it is generated by the controller card internally
- On-the-fly factor X / On-the-fly factor Y – specifies the encoder factor to be used along the related axis, the value has to be given in unit  $\mu\text{m}/\text{inc}$
- A0/A1 0% voltage – these fields can be used to define the allowed voltage range for A0 and A1 analogue outputs. Here 0% defines the output voltage which has to be set at 0% power values.
- A0/A1 100% voltage – these fields can be used to define the allowed voltage range for A0 and A1 analogue outputs. Here 0% defines the output voltage which has to be set at 100% (means full) power values.

#### MOPA:

This tab-pane contains different settings that are specific to MOPA-lasers and apply to the configuration only when a MOPA-laser was selected. Here following options can be set:

- Alarm input (12) – alarm input at the digital interface to be used for alarm-signal 12 of the laser. When this value is set to a value different than “unused”, the alarm state of the related pin at the laser is watched. As long as this pin does not signal “ready”, marking is impossible and an error message is shown in the mark dialogue. It is recommended to configure DIn4 here.
- Alarm input (11) – alarm input at the digital interface to be used for alarm-signal 11 of the laser. When this value is set to a value different than “unused”, the alarm state of the related pin at the laser is watched. As long as this pin does not signal “ready”, marking is impossible and an error message is shown in the mark dialogue. It is recommended to configure DIn5 here.
- Alarm input (16) – alarm input at the digital interface to be used for alarm-signal 16 of the laser. When this value is set to a value different than “unused”, the alarm state of the related pin at the laser is watched. As long as this pin does not signal “ready”, marking is impossible and an error message is shown in the mark dialogue. It is recommended to configure DIn6 here.
- Alarm input (21) – alarm input at the digital interface to be used for alarm-signal 21 of the laser. When this value is set to a value different than “unused”, the alarm state of the related pin at the laser is watched. As long as this pin does not signal “ready”, marking is impossible and an error message is shown in the mark dialogue. It is recommended to configure DIn7 here.
- Turn off MO during jumps – this parameter is relevant for lasers with an unusual high leakage that can't be turned off by LaserGate completely. When this option is set, the main oscillator is turned off before a jump is performed in order to minimize that leakage during scanner movement with no emission
- Set frequency before MO – when this parameter is set, the frequency at LaserA is set before the main oscillator is turned on. This may be necessary for some specific MOPA-lasers in order to let



them synchronise to the modulation frequency properly. When this option is set, the following parameter can be configured too:

- Delay before MO – here a value in unit msec can be specified which adds a delay after a new frequency has been set and before the main oscillator is turned on again. This may be necessary for some specific MOPA-lasers that react slow on change of the modulation frequency
- Delay after MO – when a value in unit msec is given here, a delay is added every time after the main oscillator was turned on. The main oscillator typically needs some settling time until the laser can accept further hardware signals. The correct delay time to be added here depends on the laser type. The default value of 10 msec should work with all major laser types.
- Delay before MO off – when a value in unit msec is given here, a delay is added every time before the main oscillator is turned off. This is a value specific to the used type of laser. The majority of laser models shall not need this delay, so this value typically can be set to 0.
- PWM control – typically a MOPA laser runs with a fixed duty cycle of 50%. But for some specific types of lasers it is possible to adjust the pulse-width too. Here a proper mode can be selected which defines what method of setting the pulse width has to be used. The value to be select here depends on the used laser type and its capabilities, so the preferred mode has to be chosen dependent on the used laser. As soon as this option is no longer set to “unused”, the pulse-width-parameter or the APD-index-parameter in pen settings becomes active and can be adjusted for every pen. Following methods of setting the pulse-width are supported:

PWM Type	Remarks
PWM value via LaserB, LP8_1, LP8_2	When this option is set, the pulse-width is transferred to the laser using a serial control information. This serial data transmission is done via the specified laser interface pins. In this mode the pulse-width is generated by the laser.
APD-index via DOut2, DOut3, DOut3	When this option is selected, an APD-index value that specifies a pulse-width is transferred to the laser using a serial control information. This serial data transmission is done via the specified digital interface pins. These pins have to be connected to the laser in this mode and are not allowed to be used for anything else. In this mode the pulse-width is generated by the laser.
PWM signal via LaserA	The pulse-width of the LaserA-signal is changed according to the current pen-settings. In this mode the scanner controller influences the pulse-width directly, it is not generated by the laser.

All following tab-panes contain laser-specific definitions that are used depending on the selected laser type. For information about these parameters and their usage please refer to the manuals and specifications of the used lasers.

This plug-in is located in flowplugins/libio\_e1803.

#### Makeblock XY-Plotter:

Warning: This plug-in is designed to control laser or other dangerous equipment which may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured.

This plug-in can be used to control a Makeblock XY-plotter by sending appropriate G-Code commands to it via serial interface. It is independent from the tool used at this plotter, which can be a laser or any other hardware that can be toggled by straight on/off signals. The plug-in provides a basic configuration panel where the serial interface parameters can be set according to the specifications of the laser to be connected.

Within a second panel the default mark-speed to be used can be configured.

This plug-in is located in flowplugins/libio\_makeblock\_xy.

Printer driver controlled:

**Warning:** This plug-in is designed to control laser equipment which may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured. Beside of that some laser equipment can be damaged in case it is controlled with wrong signals. Thus it is highly recommended to check the output generated by this plug in using e.g. an oscilloscope to avoid problems caused by wrong configurations. This should be done prior to putting a system into operation for the first time and prior to putting a system into operation after a software update was installed.

This plug-in can be used for all scanner controllers that can be accessed via a standard system printer driver. So within the configuration dialogue no laser/scanner related parameters can be set but values for line width, orientation and others:

- Field Left Position / Field Upper Position – defines the upper left coordinate of the field that has to be used by the scanner
- Field Size – the total size of the field that is used by the scanner, this parameter together with the left and upper position specifies the coordinates input vectors have to be located within, in case there are input data which define a position outside this field they will not be marked
- Swap X and Y – swaps the input X and Y coordinate to exchange X and Y axis values of the scanner
- Mirror X / Mirror Y – mirror the input coordinates

This plug-in is located in flowplugins/libio\_sc\_printer.

Raylase(R) SP-ICE2:

**Warning:** This plug-in is designed to control laser equipment which may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured. Beside of that some laser equipment can be damaged in case it is controlled with wrong signals. Thus it is highly recommended to check the output generated by this plug in using e.g. an oscilloscope to avoid problems caused by wrong configurations. This should be done prior to putting a system into operation for the first time and prior to putting a system into operation after a software update was installed.

This plug-in can be used to access the Raylase SP-ICE2 scanner controller card for controlling a scan head and a connected laser in real-time. Within the configuration dialogue the plug-in and the scanner controller card have to be configured according to the connected devices.

Basic:

- IP – the IP of the board to be controlled by this plug-in
- Port – the port number of the board
- Laser type – the type of the connected laser, the field that was selected here directly corresponds to the laser tab panes where some additional, laser-specific settings can be done; for a detailed description of the available options please refer to the manual of the SP-ICE2 and to the manuals of the laser device vendor
- Power output – the laser output power can be controlled via an additional output; the kind of output to be used can be defined here
- Number of Axes – specifies if the scanner controller has to operate in 2D or 3D mode, this option depends on the capabilities of the used scanner controller card
- Correction file – the .gcd correction file to be used for operation (provided by the vendor of the scan head)
- Stand By Frequency – frequency the laser has to be driven with in standby mode

- Stand By Length – pulse length the laser has to be driven with in standby mode
- Field Left Position / Field Upper Position – defines the upper left coordinate of the field that has to be used by the scanner
- Field Size – the total size of the field that is used by the scanner, this parameter together with the left and upper position specifies the coordinates input vectors have to be located within, in case there are input data which define a position outside this field they will not be marked
- Swap X and Y – swaps the input X and Y coordinate to exchange X and Y axis values of the scanner
- Mirror X / Mirror Y – mirror the input coordinates
- On-the-fly factor X / On-the-fly factor Y – specifies the factor in X or Y direction for marking on-the-fly operation; this option depends on the capabilities of the used scanner controller card

All following tab-panes contain laser-specific definitions that are used depending on the selected laser type. For information about these parameters and their usage please refer to the manuals and specifications of the used lasers.

This plug-in is located in flowplugins/libio\_spice2.

#### SCANLAB(R) RTC ScanAlone:

**Warning:** This plug-in is designed to control laser equipment which may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured. Beside of that some laser equipment can be damaged in case it is controlled with wrong signals. Thus it is highly recommended to check the output generated by this plug in using e.g. an oscilloscope to avoid problems caused by wrong configurations. This should be done prior to putting a system into operation for the first time and whenever a software update was installed.

This plug-in can be used to access the SCANLAB RTC ScanAlone scanner controller card for controlling a scan head and a connected laser in real-time. This plug-in supports the direct operation mode only, no writing of laser marking data onto the card or operating in stand-alone mode is supported. Within the configuration dialogue the plug-in and the scanner controller card have to be configured according to the connected devices.

Basic:

- Laser type – the type of the connected laser, the field that was selected here directly corresponds to the laser tab panes where some additional, laser-specific settings can be done; for a detailed description of the available options please refer to the manual of the RTC ScanAlone and to the manuals of the laser device vendor
- Power output – the laser output power can be controlled via an additional output; the kind of output to be used can be defined here
- Number of Axes – specifies if the scanner controller has to operate in 2D or 3D mode, this option depends on the capabilities of the used scanner controller card
- Correction file – the .ctb correction file to be used for operation (provided by the vendor of the scanhead)
- Stand By Frequency – frequency the laser has to be driven with in standby mode
- Stand By Length – pulse length the laser has to be driven with in standby mode
- Field Left Position / Field Upper Position – defines the upper left coordinate of the field that has to be used by the scanner
- Field Size – the total size of the field that is used by the scanner, this parameter together with the left and upper position specifies the coordinates input vectors have to be located within, in case there are input data which define a position outside this field they will not be marked
- Swap X and Y – swaps the input X and Y coordinate to exchange X and Y axis values of the scanner
- Mirror X / Mirror Y – mirror the input coordinates

- On-the-fly factor X / On-the-fly factor Y – specifies the factor in X or Y direction for marking on-the-fly operation; this option depends on the capabilities of the used scanner controller card

All following tab-panes contain laser-specific definitions that are used depending on the selected laser type. For information about these parameters and their usage please refer to the manuals and specifications of the used lasers.

This plug-in is located in flowplugins/libio\_rtcscanalone.

#### SCANLAB(R) RTC4:

**Warning:** This plug-in is designed to control laser equipment which may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured. Beside of that some laser equipment can be damaged in case it is controlled with wrong signals. Thus it is highly recommended to check the output generated by this plug in using e.g. an oscilloscope to avoid problems caused by wrong configurations. This should be done prior to putting a system into operation for the first time and prior to putting a system into operation after a software update was installed.

This plug-in can be used to access the SCANLAB RTC4 scanner controller card for controlling a scan head and a connected laser in real-time. Within the configuration dialogue the plug-in and the scanner controller card have to be configured according to the connected devices.

Basic:

- Use Board Number – in case more than only one board is used within the same host computer this value specifies which of the boards has to be used by this plug-in
- Laser type – the type of the connected laser, the field that was selected here directly corresponds to the laser tab panes where some additional, laser-specific settings can be done; for a detailed description of the available options please refer to the manual of the RTC4 and to the manuals of the laser device vendor
- Power output – the laser output power can be controlled via an additional output; the kind of output to be used can be defined here
- Number of Axes – specifies if the scanner controller has to operate in 2D or 3D mode, this option depends on the capabilities of the used scanner controller card
- Correction file – the .ctb correction file to be used for operation (provided by the vendor of the scan head)
- Firmware file – the .hex firmware file to be used to operate the scanner controller card (provided by SCANLAB)
- Stand By Frequency – frequency the laser has to be driven with in standby mode
- Stand By Length – pulse length the laser has to be driven with in standby mode
- Field Left Position / Field Upper Position – defines the upper left coordinate of the field that has to be used by the scanner
- Field Size – the total size of the field that is used by the scanner, this parameter together with the left and upper position specifies the coordinates input vectors have to be located within, in case there are input data which define a position outside this field they will not be marked
- Swap X and Y – swaps the input X and Y coordinate to exchange X and Y axis values of the scanner
- Mirror X / Mirror Y – mirror the input coordinates
- On-the-fly factor X / On-the-fly factor Y – specifies the factor in X or Y direction for marking on-the-fly operation; this option depends on the capabilities of the used scanner controller card

All following tab-panes contain laser-specific definitions that are used depending on the selected laser type. For information about these parameters and their usage please refer to the manuals and specifications of the used lasers.

This plug-in is located in flowplugins/libio\_rtc4.

## SCANLAB(R) RTC5:

**Warning:** This plug-in is designed to control laser equipment which may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured. Beside of that some laser equipment can be damaged in case it is controlled with wrong signals. Thus it is highly recommended to check the output generated by this plug in using e.g. an oscilloscope to avoid problems caused by wrong configurations. This should be done prior to putting a system into operation for the first time and prior to putting a system into operation after a software update was installed.

This plug-in can be used to access the SCANLAB RTC5 scanner controller card for controlling a scan head and a connected laser in real-time. Within the configuration dialogue the plug-in and the scanner controller card have to be configured according to the connected devices.

Basic:

- Use Board Number – in case more than only one board is used within the same host computer this value specifies which of the boards has to be used by this plug-in
- Laser type – the type of the connected laser, the field that was selected here directly corresponds to the laser tab panes where some additional, laser-specific settings can be done; for a detailed description of the available options please refer to the manual of the RTC5 and to the manuals of the laser device vendor
- Power output – the laser output power can be controlled via an additional output; the kind of output to be used can be defined here
- Number of Axes – specifies if the scanner controller has to operate in 2D or 3D mode, this option depends on the capabilities of the used scanner controller card
- Correction file – the .ct5 correction file to be used for operation (provided by the vendor of the scan head)
- Firmware data directory – the directory that contains the firmware files to be used to operate the scanner controller card (provided by SCANLAB)
- Stand By Frequency – frequency the laser has to be driven with in standby mode
- Stand By Length – pulse length the laser has to be driven with in standby mode
- Field Left Position / Field Upper Position – defines the upper left coordinate of the field that has to be used by the scanner
- Field Size – the total size of the field that is used by the scanner, this parameter together with the left and upper position specifies the coordinates input vectors have to be located within, in case there are input data which define a position outside this field they will not be marked
- Swap X and Y – swaps the input X and Y coordinate to exchange X and Y axis values of the scanner
- Mirror X / Mirror Y – mirror the input coordinates
- On-the-fly factor X / On-the-fly factor Y – specifies the factor in X or Y direction for marking on-the-fly operation; this option depends on the capabilities of the used scanner controller card

All following tab-panes contain laser-specific definitions that are used depending on the selected laser type. For information about these parameters and their usage please refer to the manuals and specifications of the used lasers.

This plug-in is located in flowplugins/libio\_rtc5.

## SCANLAB(R) RTC6:

**Warning:** This plug-in is designed to control laser equipment which may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured. Beside of that some laser equipment can be damaged in case it is controlled with wrong signals. Thus it is highly recommended to check the output generated by this plug in using e.g. an oscilloscope to avoid problems caused by wrong

configurations. This should be done prior to putting a system into operation for the first time and prior to putting a system into operation after a software update was installed.

This plug-in can be used to access the SCANLAB RTC6 scanner controller card for controlling a scan head and a connected laser in real-time. Within the configuration dialogue the plug-in and the scanner controller card have to be configured according to the connected devices.

Basic:

- Use Board Number – in case more than only one board is used within the same host computer this value specifies which of the boards has to be used by this plug-in
- Laser type – the type of the connected laser, the field that was selected here directly corresponds to the laser tab panes where some additional, laser-specific settings can be done; for a detailed description of the available options please refer to the manual of the RTC6 and to the manuals of the laser device vendor
- Power output – the laser output power can be controlled via an additional output; the kind of output to be used can be defined here
- Number of Axes – specifies if the scanner controller has to operate in 2D or 3D mode, this option depends on the capabilities of the used scanner controller card
- Correction file – the .ct5 correction file to be used for operation (provided by the vendor of the scan head)
- Firmware data directory – the directory that contains the firmware files to be used to operate the scanner controller card (provided by SCANLAB)
- Stand By Frequency – frequency the laser has to be driven with in standby mode
- Stand By Length – pulse length the laser has to be driven with in standby mode
- Field Left Position / Field Upper Position – defines the upper left coordinate of the field that has to be used by the scanner
- Field Size – the total size of the field that is used by the scanner, this parameter together with the left and upper position specifies the coordinates input vectors have to be located within, in case there are input data which define a position outside this field they will not be marked
- Swap X and Y – swaps the input X and Y coordinate to exchange X and Y axis values of the scanner
- Mirror X / Mirror Y – mirror the input coordinates
- On-the-fly factor X / On-the-fly factor Y – specifies the factor in X or Y direction for marking on-the-fly operation; this option depends on the capabilities of the used scanner controller card

All following tab-panes contain laser-specific definitions that are used depending on the selected laser type. For information about these parameters and their usage please refer to the manuals and specifications of the used lasers.

This plug-in is located in flowplugins/libio\_rtc6.

SCAPS(R) USC-1/2/3:

**Warning:** This plug-in is designed to control laser equipment which may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured. Beside of that some laser equipment can be damaged in case it is controlled with wrong signals. Thus it is highly recommended to check the output generated by this plug in using e.g. an oscilloscope to avoid problems caused by wrong configurations. This should be done prior to putting a system into operation for the first time and prior to putting a system into operation after a software update was installed.

These plug-ins can be used to access the SCAPS USC-1, USC-2 and USC-3 scanner controller cards via the generic Scanner Controller Interface for controlling a scan head and a connected laser in real-time. The usage of this interface may require additional licenses provided by the vendor of the USC scanner controller cards.

Within the configuration dialogue the plug-in, the scanner controller card and the interface to the card have to be configured according to the connected devices.

Basic:

- Laser type – the type of the connected laser, the field that was selected here directly corresponds to the laser tab panes where some additional, laser-specific settings can be done; for a detailed description of the available options please refer to the SCAPS manuals and to the manuals of the laser device vendor
- Power output – the laser output power can be controlled via an additional output; the kind of output to be used can be defined here
- Number of Axes – specifies if the scanner controller has to operate in 2D or 3D mode, this option depends on the capabilities of the used scanner controller card and on the available SCAPS licenses
- Correction file – the .ucf correction file to be used for operation (provided by the vendor of the scan head or by SCAPS)
- Stand By Frequency – frequency the laser has to be driven with in standby mode
- Stand By Length – pulse length the laser has to be driven with in standby mode
- Field Left Position / Field Upper Position – defines the upper left coordinate of the field that has to be used by the scanner
- Field Size – the total size of the field that is used by the scanner, this parameter together with the left and upper position specifies the coordinates input vectors have to be located within, in case there are input data which define a position outside this field they will not be marked
- Swap X and Y – swaps the input X and Y coordinate to exchange X and Y axis values of the scanner
- Mirror X / Mirror Y – mirror the input coordinates
- On-the-fly factor X / On-the-fly factor Y – specifies the factor in X or Y direction for marking on-the-fly operation; this option depends on the capabilities of the used scanner controller card and on the available SCAPS licenses

All following tab-panes contain laser-specific definitions that are used depending on the selected laser type. For information about these parameters and their usage please refer to the manuals and specifications of the used lasers.

These plug-ins are located in flowplugins/libio\_samsci, flowplugins/libio\_samsci2 and flowplugins/libio\_samsci3

Sintec(R) ETH6608:

**Warning:** This plug-in is designed to control laser equipment which may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured. Beside of that some laser equipment can be damaged in case it is controlled with wrong signals. Thus it is highly recommended to check the output generated by this plug in using e.g. an oscilloscope to avoid problems caused by wrong configurations. This should be done prior to putting a system into operation for the first time and prior to putting a system into operation after a software update was installed.

This plug-in can be used to access the Sintec ETH6608 scanner controller card for controlling a scan head and a connected laser in real-time. Within the configuration dialogue the plug-in and the scanner controller card have to be configured according to the connected devices.

Basic:

- Board IP – the IP of the board that has to be accessed by this plug-in
- Laser type – the type of the connected laser, the field that was selected here directly corresponds to the laser tab panes where some additional, laser-specific settings can be done; for a detailed description of the available options please refer to the manual of the ETH6608 and to the manuals of the laser device vendor

- Scanner Output – the output that has to be used for accessing the scanner, here it can be chosen between analogue output and XY2/100 interface; this has to be set to the output the scanner is connected with
- Power output – the laser output power can be controlled via an additional output; the kind of output to be used can be defined here
- Number of Axes – specifies if the scanner controller has to operate in 2D or 3D mode, this option depends on the capabilities of the used scanner controller card
- Correction file – the .bco, .ctb, .ct5, .gcd or .ucf correction file to be used for operation (provided by the vendor of the scan head)
- Stand By Frequency – frequency the laser has to be driven with in standby mode
- Stand By Length – pulse length the laser has to be driven with in standby mode
- Field Left Position / Field Upper Position – defines the upper left coordinate of the field that has to be used by the scanner
- Field Size – the total size of the field that is used by the scanner, this parameter together with the left and upper position specifies the coordinates input vectors have to be located within, in case there are input data which define a position outside this field they will not be marked
- Swap X and Y – swaps the input X and Y coordinate to exchange X and Y axis values of the scanner
- Mirror X / Mirror Y – mirror the input coordinates
- On-the-fly speed X / On-the-fly speed Y – specifies the used static speed in X or Y direction for marking on-the-fly operation

All following tab-panes contain laser-specific definitions that are used depending on the selected laser type. For information about these parameters and their usage please refer to the manuals and specifications of the used lasers.

This plug-in is located in flowplugins/libio\_eth6608.

## 5.5.2 Motion Controller Configuration

HALaser E1701M Stepper Motor Controller:

Using this plug-in up to four axes of E1701M stepper motor controller by HALaser Systems can be accessed. These axes can be moved independently or synchronously via the control-input. In second case it is possible to perform flat, linear X/Y movements or three-dimensional X/Y/Z movements. It can be used e.g. together with an XY-Table and a tool for machining purposes. The controller can be accessed either via USB serial interface or via Ethernet.

Within the configuration panels of the plug-in there are two major set-up possibilities. The first panel provides general settings and parameters that apply to the plug-in and controller globally:

- IP or Serial Interface - here it has to be configured how and where the controller is accessible. In case of Ethernet connection the IP (in style aaa.bbb.ccc.ddd) has to be entered here, in case of USB-connection the name of the serial interface has to be given ("COMx" for Windows, "/dev/ttyACMx" for Linux where "x" is the number of the serial interface the controller was connected with)
- Ethernet Password - this parameter has to be given only in case Ethernet connection is used; here a password can be specified in order to identify connection to a controller. The same password has to be set within the related E1701M motion controller. When password specified here and password configured at connected controller do not fit, connection is closed and plug-in is not able to use this controller.
- Swap X and Y - this option is valid only in case the plug-in is fed with coordinates and not with separate position values for each axis, when it is set, X and Y coordinate are exchanged. When plug-in is used out of BeamConstruct, this option is inoperable



The settings within the axis-panels apply for one specific axis only. Here different settings for each axis can be given:

- **Enable Axis** – an axis is used and can be moved by this plug-in only in case this checkbox is set. When plug-in has to be used in coordinate mode (access via Control-input where XY or XYZ position coordinates are sent to it instead of separate axis position data) axis 1 and 2 have to be enabled for 2D coordinates (X and Y) and axis 1, 2 and 3 have to be enabled for 3D coordinates (X, Y and Z)
- **Axis Mode** – an axis can be operated in two modes: planar or radial. In planar mode it moves over a distance (using unit mm), in angular mode it moves to an angle (in unit degrees)
- **Resolution Factor** – this value has to be used to specify conversion factor from shown position value (mm in planar mode, degrees in angular mode) to incremental position of the used motor (depending on its incremental resolution, step-configuration of used driver and used gearbox)
- **Low Limit** – the lower limit (in unit mm) is a soft limit specifying the minimum position the axis is allowed to move to in planar mode; in case of angular mode this value is ignored.  
PLEASE NOTE: when an axis was not referenced yet, this limit is useless since the real position of axis is not known to E1701M controller. So as very first an axis always should be referenced!
- **High Limit** – the higher limit (in unit mm) is a soft limit specifying the maximum position the axis is allowed to move to in planar mode; in case of angular mode this value is ignored.  
PLEASE NOTE: when an axis was not referenced yet, this limit is useless since the real position of axis is not known. So as very first an axis always should be referenced!
- **Maximum Speed** – this parameter can be used to limit maximum speed an axis can move with (in unit mm/sec for planar mode and deg/sec in angular mode). Independent from which speed value is specified during operation, this limit is never exceeded
- **Acceleration Mode** – specifies the shape of the ramp used during acceleration, here it can be chosen between linear (smooth acceleration), exponential (strong but not so smooth mode which may overshoot at end of acceleration phase) or s-shaped acceleration (very smooth but not very fast accelerating)
- **Acceleration** – factor that specifies how strong acceleration is for chosen acceleration mode
- **Deceleration Mode** – specifies the shape of the ramp used during deceleration, here it can be chosen between linear (smooth deceleration), exponential (strong but not so smooth mode which may lose increments at beginning of deceleration phase) or s-shaped deceleration (very smooth but not very fast)
- **Deceleration** – factor that specifies how strong deceleration is for chosen deceleration mode
- **Stop Deceleration** – factor that specifies strength of deceleration for all stop events (stop by limit switch, reference switch or by stop-command)
- **Invert Direction** – all axis use a "positive" and "negative" movement direction which are used to describe the motion direction. When this option is set, "positive" and "negative" are exchanged which may be useful to make UI-layout/appearance of these directions more convenient to user
- **Reference Mode** – after first start-up or when a motion error occurred where increments may got lost, referencing is required for an axis. To do this, a reference switch is required at a defined mounting position (refer to related parameter below). Using reference mode parameter a sequence of movements in relation to this reference switch can be chosen that has to be performed on referencing.
- **Reference Timeout** – here a value in unit seconds can be specified. When referencing could not be finished after this time has elapsed, it will be cancelled since it seems axis is not able to find reference switch and/or movement is not possible for some reasons
- **Reference Signal Input** – here an input can be specified that is used as switch defining the referencing position
- **Invert input logic** – inverts the logic of the reference input to react on LOW level instead of HIGH
- **Reference Speed 1/2/3** – referencing is done in up to three different steps (depending on used reference mode). Speeds for these steps should be slow in order to not overshoot when reference position is reached. In case of more than one referencing step, these speed values should decrease

for every step. Using these three parameters reference speed values can be specified. When a speed is given for a step that is not used in chosen reference mode, this value is ignored.

- Referenced Position – after referencing was finished successfully, the current axis position is set to the position value specified here
- Limit Switch 1/2 Input – beside the soft limits it is possible to define hardware-controlled position limits via limit switches. When such limit switches are used and connected to E1701M controller, the inputs they are connected with can be specified here. Whenever one of these switches sends a HIGH signal, current movement is stopped using stop-deceleration
- Invert input logic – inverts the logic of the limit switch input to react on LOW level instead of HIGH
- Auto-leave Limit Switch – when a limit switch is hit, normally no more motions are allowed until this switch is left. When this option is set and an axis hits a limit switch, it automatically moves back until this switch is left.  
PLEASE NOTE: when this option is set, a possibly unexpected motion may start in opposite direction!
- Encoder – E1701M controller board supports up to two external quadrature encoders that provide position information. When such an encoder is connected to decoder input 0 (digital inputs 0 and 1) or decoder input 1 (digital inputs 2 and 3) these inputs can be enabled here for an axis. When an external encoder is activated newly, the axis position on start-up is set to 0. To specify a different position, reference mode "No referencing" has to be chosen, a referencing position has to be set and referencing has to be performed. This does not cause any referencing operation but immediately sets the current, encoder-based position to the one specified as "Referenced Position".  
Alternatively a standard referencing operation as described above can be performed too.  
PLEASE NOTE: an encoder can be used for one axis only, when the same decoder input is configured for more than one axis, the results are undefined.
- Encoder Factor – this factor is used for calculation of axis incremental position, here the ratio between encoder pulses and motor increments has to be specified

This plug-in is located in flowplugins/libio\_e1701m\_stepper.

HALaser Systems E1803D Motion (Intelli-IO controlled):

Using this plug-in a stepper motor can be controlled using the Intelli-IO extension of the E1803D scanner controller card. This plug-in accesses the Intelli-IO and operates it in motion mode.

The plug-in is able to control up to four axes via two output signals "Step" and "Direction". These axes will always combined, two or more axes will move at the same time using a common motion speed which does not exceed the speed values set for each of the axes. Combined movement means, it is done in a way where all axes start and stop at the same time.

As global parameter only the "Step pulsewidth" can be set. This value influences the length of a single pulse emitted at the step-outputs of the Intelli-IO extension. Normally this value can be left at the default value of 10 µsec. Smaller values does not make any sense here. And only in case of some slow or low-quality stepper motor drivers it may be necessary to set larger values here in order to let them react on a single pulse properly. Please note: the larger the step pulsewidth becomes, the lower is the maximum step frequency the Intelli-IO board can emit and the lower is the maximum speed an axis can be moved with!

Within the axis-related configuration panels parameters can be set for each of the axes:

- Enable axis – turns on the axis to be used for motion operations
- Axis Mode – specifies the operational mode of the drive, here out of the two options planar (expects linear input data) and rotational (expects angular input values in unit degree) can be chosen
- Factor – specifies a factor that is used to calculate a metric value in millimetres or a rotational value in degrees out of the drive-internal position information that is given in unit "increments", the exact factor depends on your hardware set-up and the gear you are using
- Low Limit – specifies the lower limit in unit millimetres where the axis is allowed to move to at minimum

- High Limit – specifies the upper limit in unit millimetres where the drive is allowed to move to at maximum
  - Maximum speed – defines the maximum speed an axis is allowed to be operated with. Please note: in case of a modified, larger step pulsewidth (please refer to global parameter described above) this value should reflect the lower maximum speed an axis is able to move with
  - Acceleration – a value that specifies how smooth the drive should start and stop all its movements
  - Invert Direction – inverts the direction output / the movement direction for this axis
  - Reference Mode – here it can be specified if a home/referencing operation has to be done or not and if yes which method of referencing has to be used; the chosen mode defines using which direction the software shall search for the reference switch and using which direction the switch shall be left afterwards
  - Reference Timeout – a timeout value in seconds after what a referencing operation will be cancelled no matter if the reference switch was hit or the referencing sequence was completed
  - Reference Speed 1 – the speed that has to be used to search for the reference switch in first referencing cycle, for a detailed description of the referencing procedure, please refer below
  - Reference Speed 2 – the speed that has to be used to search for the reference switch in second referencing cycle; this value should be smaller than the “Reference Speed 1”.
- A referencing cycle of the Intelli-IO extension consist of the following steps for an axis:
1. move to reference switch using direction selected via “Reference Mode” option and with “Reference Speed 1”
  2. stop axis as soon as switch was hit
  3. leave the reference switch by travelling 1 mm in opposite direction; Please note: when this distance is not enough to leave the switch, set the “Reference Speed 1” to a lower value that ensures the switch is not passed by more than 1 mm
  4. move to reference switch again using direction selected via “Reference Mode” option and with “Reference Speed 2”
  5. leave the reference switch by travelling 1 mm in opposite direction again; Please note: when this distance is not enough to leave the switch, set the “Reference Speed 2” to a lower value!
  6. Stop axis, referencing is done now and the current position is set to “Referenced Position” as defined below
- Referenced Position – the metric position (in unit millimetres) that has to be set at the referenced position

#### Schneider/IMS MDrive+:

This plug-in is able to control up to three axes of a Schneider/IMS MDrive+; there is also a macro available that encapsulates the low level functionalities described here. These axes can be moved independently or synchronously via the control-input to perform flat, linear X/Y movements (or three-dimensional X/Y/Z movements). This mode can be used e.g. together with an XY-Table and a tool for machining purposes.

This drive can be controlled via a serial interface where all axes are connected to this line. Independent from the fact if one or more axes are used, all of the used drives have to be pre-configured in the following way before operating them within a BeamConstruct environment. Following the configuration steps are described together with the commands that have to be sent to the drive to do this configuration:

1. A unique device identifier has to be set for each axis (DN=“X” where “X” is the identifier).
2. The echo mode has to be set to mode “2” (EM=2).
3. The party mode has to be enabled (PY=1).
4. These parameters have to be stored to the drives permanently (XS^J).
5. When the homing/referencing functionality has to be used, the input S1 of the drive has to be connected to the home switch.

When this configuration was done successfully, the drives can be set up within the configuration panel of the plug-in. First of all the data of the serial interface have to be defined. Here the correct port name and the interface settings have to be given in order to establish the communication to the drive. Next for every used axis several parameters have to be given:

- Axis Mode – specifies the operational mode of the drive, here out of the two options planar (expects linear input data) and rotational (expects angular input values in unit degree) can be chosen
- Factor - specifies a factor that is used to calculate a metric value in millimetres or an angular value in degrees out of the drive-internal position information that is given in unit "increments", the exact factor depends on your hardware set-up and the used gear
- Low Limit - specifies the lower limit in unit millimetres where the axis is allowed to move to at minimum
- High Limit - specifies the upper limit in unit millimetres where the drive is allowed to move to at maximum
- Axis Name - is the device name that was stored within the drive in order to identify it
- Axis Home Timeout - is a time-out value in seconds that described after what time referencing has to be stopped when no home switch can be found

This plug-in is located in flowplugins/libio\_ims\_mdrive2.

#### IO Sync Motor Trigger (Scannercard):

This plug-in is not a motor driver in common sense. Instead of submitting motion and position information, it sends a digital signal to a connected device which then can be used to trigger external equipment. This digital signal stays at HIGH until the external device responds with an other signal. As soon as this signal is received, the software assumes the motion is completed and continues with other operations. This plug-in sends and receives the digital signals via a connected scanner controller card and its digital IOs.

The plug-in is able to control up to three axes via these digital on/off signals. These axes will operate sequentially, two or more axes will not be triggered at the same time.

As global parameter only the digital output port and the digital input port of the connected scanner controller card have to be chosen. Here is up to the user to select IO ports that really exist for the used scanner controller, in case an IO is specified that is not available, no data can be send to it or no input information can be received. Which bit of the selected IO's is used for which purpose and which axis, can be specified within the axis-related configuration panels:

- Motion start output – the output bit at the previously chosen output port type that is used to send the command to an external device to start a motion operation. This output stays at HIGH until the external device signals the motion operation was completed
- Motion done input – the input bit at the previously chosen input port where the external device signals the end of a motion operation
- Reference start output – the output bit at the previously chosen output port type that is used to send the command to an external device to start a referencing operation. This output stays at HIGH until the external device signals the referencing operation was completed
- Reference Timeout – a timeout value in seconds after what a referencing operation will be cancelled no matter if the connected device signalled a "motion done" or not
- Referenced Position – the metric position (in unit millimetres) that has to be set at the referenced position

This plug-in is located in flowplugins/libio\_scanctrl\_iosync.

#### Stepper Motor Driver (Scannercard):

Using this plug-in a stepper motor can be controlled via a connected scanner controller card. This plug-in does not control such a card directly but it generates output data that can be handled by a scanner controller card.

The plug-in is able to control up to four axes via two output signals "Step" and "Direction". These axes will operate sequentially, two or more axes will not move at the same time.

As global parameter only the digital output port and the digital input port of the connected scanner controller card have to be chosen. Here it is up to the user to select IO ports that really exist for the used scanner controller, in case an IO is specified that does not exist, no data can be sent to it or no input information can be received. Which bit of the selected IO's is used for which purpose and which axis can be specified within the axis-related configuration panels:

- Axis Mode – specifies the operational mode of the drive, here out of the two options planar (expects linear input data) and rotational (expects angular input values in unit degree) can be chosen
- Factor – specifies a factor that is used to calculate a metric value in millimetres or a rotational value in degrees out of the drive-internal position information that is given in unit "increments", the exact factor depends on your hardware set-up and the gear you are using
- Low Limit – specifies the lower limit in unit millimetres where the axis is allowed to move to at minimum
- High Limit – specifies the upper limit in unit millimetres where the drive is allowed to move to at maximum
- Acceleration – a value that specifies how smooth the drive should start its movements
- Deceleration – a value that specifies how smooth the drive should stop its movements
- Step Signal Output – the output bit at the previously chosen output port type that is used for the pulsed step signal; here it has to be made sure that an output is used only once, overlapping mappings would cause problems during operation
- Direction Signal Output – the output bit at the output port that was chosen in general configuration panel, this output bit is used to define the movement direction of the axis; here it has to be made sure that an output is used only once, overlapping mappings would cause problems during operation
- Invert Direction – inverts the direction output / the movement direction for this axis
- Invert input logic – inverts the logic of the limit switch input to react on LOW level instead of HIGH
- Lazy input switch – when stepper motors are operated via this software control and digital in/outputs of a controller card instead of a native motion controller, all operations are done somewhat slower because of the non-realtime behaviour of the operating system BeamConstruct is running at. Thus the whole system can't react as fast as an embedded controller e.g. when a reference switch was hit and therefore may not find it successfully. In this case the user has the possibility to slow down the reference speed and/or to set this option in order to tolerate more variances during referencing. To still get an accurate reference position under these conditions, the "Reference Speed (Leave)" needs to be set to lower values.
- Reference Mode – here it can be specified if a home/referencing operation has to be done or not and if yes which method of referencing has to be used; the chosen mode defines using which direction the software shall search for the reference switch and using which direction the switch shall be left afterwards
- Reference Signal Input – the input bit at the previously chosen input port where the reference switch is connected to
- Reference Speed (Enter) – the speed that has to be used to search for the reference switch
- Reference Speed (Leave) – the speed that has to be used to leave the reference switch once it was found; the correct referenced position is that position, where the reference switch just turns off; this part of referencing has a direct influence on the accuracy on the resulting position, the lower it is, the more exact the position will be. Please also refer to option "Lazy input switch" described above.
- Reference Timeout – a timeout value in seconds after what a referencing operation will be cancelled no matter if the reference switch was hit or the referencing sequence was completed

- Referenced Position – the metric position (in unit millimetres) that has to be set at the referenced position

This plug-in is located in flowplugins/libio\_scanctrl\_stepper.

### 5.5.3 Image Capture Configuration

Capture Image:

Captures an image from a video device attached to the operating system. Within the plug-ins configuration panel the device number to capture the data from can be configured. Beside of that it is possible to specify following values that influence the resulting image:

- Input poll cycle time: this value specifies how often a new image has to be captured from the input device. Small values result in faster updates which cause a more liquid displaying of images but also result in a higher load on the computer
- Scale to image width – the image is scaled to the given width (in unit pixels)
- Scale to image height – the image is scaled to the given height (in unit pixels)
- Don't scale – when this checkbox is set, the two previous values are ignored and the resulting image uses the largest resolution the capture device supports
- Brightness, Contrast, Gamma – changes the brightness, gamma a contrast of the captured image
- R, G, B – changed the red, green and blue image components; these values are applied also in case the image is converted to greyscale, here the colour correction is applied before the conversion to grey takes place
- Convert to greyscale – the captured image is converted to an image with 256 shades of grey (a luminance-based conversion is done here)
- Mirror X/Y – the image is mirrored along the X or Y axis
- Rotate: the image can be rotated by 90, 180 or 270 degrees
- Show test pattern – this is an option which can be used for testing, it disables image capture but shows a fixed, predefined testpattern instead
- Test pattern image – this option corresponds to the preceding one, when a path to an existing image file is given here, instead of the predefined test-pattern this image is shown
- Show noise – this option is a special variant of the preceding two ones; here also no image capture is done any more, but instead of a defined, static test-pattern some noise is produced here which changes on every image update

This plug-in is located in flowplugins/libio\_grabimg

IP Camera:

Captures an image from a video device which is accessible via TCP/IP (HTTP). Within the plug-ins configuration panel the URL where the image is accessible has to be set in style "<http://ip/path>". The plug-in then cyclically fetches the current image from this URL. Beside of that it is possible to specify following values that influence the resulting image:

- Input poll cycle time: this value specifies how often a new image has to be fetched from the given URL. Small values result in faster updates which cause a more liquid displaying of images but also result in a higher load on the computer
- Scale to image width – the image is scaled to the given width (in unit pixels)
- Scale to image height – the image is scaled to the given height (in unit pixels)
- Don't scale – when this checkbox is set, the two previous values are ignored and the resulting image uses the largest resolution the capture device supports

- Brightness, Contrast, Gamma – changes the brightness, gamma and contrast of the captured image
- R, G, B – changed the red, green and blue image components; these values are applied also in case the image is converted to greyscale, here the colour correction is applied before the conversion to grey takes place
- Convert to greyscale – the captured image is converted to an image with 256 shades of grey (a luminance-based conversion is done here)
- Mirror X/Y – the image is mirrored along the X or Y axis
- Rotate: the image can be rotated by 90, 180 or 270 degrees

This plug-in is located in flowplugins/libio\_ipcamera

## 5.5.4 Additional Laser Control Configuration

Coherent(R) Avia Ethernet Laser Controller:

**Warning:** This plug-in is designed to control laser equipment which may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured.

Using this plug-in a Coherent Avia laser can be controlled via Ethernet. To access the serial interface of the target laser an Ethernet to serial converter is required. The plug-in provides a basic configuration panel where the network parameters can be set according to the configuration of the TCP/IP to serial converter. A second panel gives the possibility to enter several laser-specific parameters. By default the plug-in can be used to set fixed frequency and power that are used all the time. On “Basic” configuration panel this behaviour can be changed. There are checkboxes to influence the behaviour of the plug-in:

- Control power during process – when this checkbox is set, the power-values defined in pen parameters are sent to the plug-in before the related vector data are sent to the scanner card
- Control frequency during process – when this checkbox is set, the frequency-values defined in pen parameters are sent to the plug-in before the related vector data are sent to the scanner card

This plug-in is located in flowplugins/libio\_coherent\_avia.

Generic Laser Controller:

**Warning:** This plug-in is designed to control laser equipment which may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured.

Using this plug-in a laser can be controlled via serial interface ASCII commands. It is able to connect to a laser controller via COM-port/serial interface or TCP/IP, send predefined commands to it and receive a response from the laser. The received data can be interpreted in a minimum way. The plug-in provides a basic configuration panel where the serial interface parameters can be set according to the specifications of the laser to be connected.

As parameter “Interface” following values can be set:

- “COMx” or “/dev/ttyx” in case a local serial interface has to be used, here “x” is the number of the serial port
- “U1” in case the serial interface of a connected scanner controller card has to be used, in this case all data are routed to and from this controller card
- IP in style “aaa.bbb.ccc.ddd” in case network connection via TCP/IP has to be used

A second panel gives the possibility to enter the commands that have to be sent to the laser. Here several commands can be submitted. Every command has to be separated by specific characters that specify carriage return and/or line feed. Here for a carriage return the special character sequence “[CR]” has to be added for a line feed “[LF]” can be used. Please note: also in case only one laser command has to be sent, the CR/LF termination characters have to be set after the command. Which characters have to be used

depends on the requirements of the used laser. The “Commands” configuration panel offers following configuration possibilities:

- Response End Character – specifies which characters are expected in every response from the laser; if “none” is chosen, the plug-in tries to receive some data after every command transmission but does not check the contents
- Device Open Command – here the command can be specified that is sent to the laser when everything is initialised and when all connected devices are opened
- Device Open Delay – this is an optional delay that can be specified to let the plug-in wait after the Device Open Command is sent and before all other operations (like opening of other devices) are continued; this can be useful to let the laser start up/settle completely
- Ready for Marking Command – this command is sent to the laser whenever it has to be ready for marking, it is issued e.g. in case BeamConstruct opens the Mark dialogue
- Ready for Marking Delay – this is an optional delay that can be specified to let the plug-in wait after the Ready for Marking Command is sent and before all other operations (like start of marking operation) are continued; this can be useful to let the laser start up/settle completely
- Job Start Command – this command is sent to the laser prior to submission of first vector data to the scanner card
- Job Start Delay – this is an optional delay that can be specified to let the plug-in wait after the Job Start Command is sent and before sending of marking/vector data begins
- Laser On Command – this is a very specific command that should be used only in some very special applications, it is issued whenever the laser has to be turned on; this is done by the scanner card normally, so the user has to ensure that both commands don't overlap. Please note: Since serial communication is quite slow and not very responsive this way of turning on the laser is quite inaccurate, it should be used only when exact laser timing does not matter
- Laser On Delay – this is an optional delay that can be specified to let the plug-in wait after the Laser On Command is sent and before sending of marking/vector data begins
- Laser Off Command – this corresponds to the Laser On Command and is a very specific one that should be used only in some very special applications, it is issued whenever the laser has to be turned off; this is done by the scanner card normally, so the user has to ensure that both commands don't overlap. Please note: Since serial communication is quite slow and not very responsive this way of turning off the laser is quite inaccurate, it should be used only when exact laser timing does not matter
- Job End Command – this command is sent to the laser after submission of vector data to the scanner card has finished
- End Ready for Marking Command – this command is sent to the laser whenever it no longer needs to be ready for marking, it is issued e.g. in case BeamConstruct closes the Mark dialogue
- Device Close Command – here the command can be specified that is sent to the laser when everything is de-initialised and when all connected devices are closed; this command is invoked e.g. when the main application is closed

This plug-in is located in flowplugins/libio\_lc\_generic.

#### SPI G4 Laser RS232 Controller:

Warning: This plug-in is designed to control laser equipment which may effect a person's health or may otherwise cause damage. Prior to installation and operation compliance with all relevant safety regulations including additional hardware-controlled safety measures has to be secured.

Using this plug-in a SPI G4 laser can be controlled via serial interface ASCII commands. It is able to connect to such a laser controller via COM-port/serial interface, send predefined and freely definable commands to it and receive responses from the laser.

The plug-in provides a basic configuration panel where the serial interface parameters can be set according to the communication parameters of the lasers serial interface.



A second panel gives the possibility to specify behaviour of the plug-in and to enter custom commands that have to be sent to the laser. Every of these custom commands has to be separated by specific characters that specify carriage return and/or line feed. Here for a carriage return the special character sequence "[CR]" has to be added for a line feed "[LF]" can be used. Please note: also in case only one laser command has to be sent, the "[CR] [LF]" termination characters have to be appended.

The "Commands" configuration panel offers following configuration possibilities:

- Device Open Command – here a custom command can specified that is sent to the laser when everything is initialised and when all connected devices are opened
- Device Open Delay – this is an optional delay that can be specified to let the plug-in wait after the Device Open Command is sent and before all other operations (like opening of other devices) are continued; this can be useful to let the laser start up/settle completely
- Ready for Marking Command – this custom command is sent to the laser whenever it has to be ready for marking, it is issued e.g. in case BeamConstruct opens the Mark dialogue
- Ready for Marking Delay – this is an optional delay that can be specified to let the plug-in wait after the Ready for Marking Command is sent and before all other operations (like start of marking operation) are continued; this can be useful to let the laser start up/settle completely
- Job Start Command – this command is sent to the laser prior to submission of first vector data to the scanner card
- Job Start Delay – this is an optional delay that can be specified to let the plug-in wait after the Job Start Command is sent and before sending of marking/vector data begins
- Control power during process - when this checkbox is set, the plug-in controls the power of the laser, means it sends the related command to it automatically on every change of output power and before vector data are sent that would use this new power value
- Control waveform during process - when this checkbox is set, the plug-in controls the used waveform, means it sends the related command to it automatically on every change of waveform and before vector data are sent that would use it
- Job End Command – this optional, custom command is sent to the laser after submission of vector data to the scanner card has finished
- End Ready for Marking Command – this optional command is sent to the laser whenever it no longer needs to be ready for marking, it is issued e.g. in case BeamConstruct closes the Mark dialogue
- Device Close Command – here an optional, custom command can specified that is sent to the laser when everything is de-initialised and when all connected devices are closed; this command is invoked e.g. when the main application is closed or when a user opens the global settings dialogue in BeamConstruct

This plug-in is located in flowplugins/libio\_lc\_spi.

## 5.5.5 Z-Shifter Configuration

HALaser E1701M Focus Shifter Motor Controller:

Using this plug-in up one axis of E1701M stepper motor controller by HALaser Systems can be accessed. It is intended to be used as Z-shifter/focus shifter primarily. For independent axis control of up to four axes please check description of E1701M Stepper Motor controller.

The controller can be accessed either via USB serial interface or via Ethernet.

Within the configuration panels of the plug-in there are two major set-up possibilities. The first panel provides general settings and parameters that apply to the plug-in and controller globally:

- IP or Serial Interface - here it has to be configured how and where the controller is accessible. In case of Ethernet connection the IP (in style aaa.bbb.ccc.ddd) has to be entered here, in case of USB-connection the name of the serial interface has to be given ("COMx" for Windows, "/dev/ttyACMx" for Linux where "x" is the number of the serial interface the controller was connected with)

- Ethernet Password - this parameter has to be given only in case Ethernet connection is used; here a password can be specified in order to identify connection to a controller. The same password has to be set within the related E1701M motion controller. When password specified here and password configured at connected controller do not fit, connection is closed and plug-in is not able to use this controller.

The settings within the axis-panel applies for the Z-axis:

- Resolution Factor – this value has to be used to specify conversion factor from shown position value (mm in planar mode, degrees in angular mode) to incremental position of the used motor (depending on its incremental resolution, step-configuration of used driver and used gearbox)
- Low Limit – the lower limit (in unit mm) is a soft limit specifying the minimum position the axis is allowed to move to in planar mode; in case of angular mode this value is ignored.  
PLEASE NOTE: when an axis was not referenced yet, this limit is useless since the real position of axis is not known to E1701M controller. So as very first an axis always should be referenced!
- High Limit – the higher limit (in unit mm) is a soft limit specifying the maximum position the axis is allowed to move to in planar mode; in case of angular mode this value is ignored.  
PLEASE NOTE: when an axis was not referenced yet, this limit is useless since the real position of axis is not known. So as very first an axis always should be referenced!
- Maximum Speed – this parameter can be used to limit maximum speed an axis can move with (in unit mm/sec for planar mode and deg/sec in angular mode). Independent from which speed value is specified during operation, this limit is never exceeded
- Acceleration Mode – specifies the shape of the ramp used during acceleration, here it can be chosen between linear (smooth acceleration), exponential (strong but not so smooth mode which may overshoot at end of acceleration phase) or s-shaped acceleration (very smooth but not very fast accelerating)
- Acceleration – factor that specifies how strong acceleration is for chosen acceleration mode
- Deceleration Mode – specifies the shape of the ramp used during deceleration, here it can be chosen between linear (smooth deceleration), exponential (strong but not so smooth mode which may lose increments at beginning of deceleration phase) or s-shaped deceleration (very smooth but not very fast)
- Deceleration – factor that specifies how strong deceleration is for chosen deceleration mode
- Stop Deceleration – factor that specifies strength of deceleration for all stop events (stop by limit switch, reference switch or by stop-command)
- Invert Direction – all axis use a "positive" and "negative" movement direction which are used to describe the motion direction. When this option is set, "positive" and "negative" are exchanged which may be useful to make UI-layout/appearance of these directions more convenient to user
- Reference Mode – after first start-up or when a motion error occurred where increments may got lost, referencing is required for an axis. To do this, a reference switch is required at a defined mounting position (refer to related parameter below). Using reference mode parameter a sequence of movements in relation to this reference switch can be chosen that has to be performed on referencing.
- Reference Timeout – here a value in unit seconds can be specified. When referencing could not be finished after this time has elapsed, it will be cancelled since it seems axis is not able to find reference switch and/or movement is not possible for some reasons
- Reference Signal Input – here an input can be specified that is used as switch defining the referencing position
- Invert input logic – inverts the logic of the reference input to react on LOW level instead of HIGH
- Reference Speed 1/2/3 – referencing is done in up to three different steps (depending on used reference mode). Speeds for these steps should be slow in order to not to overshoot when reference position is reached. In case of more than one referencing step, these speed values should decrease for every step. Using these three parameters reference speed values can be specified. When a speed is given for a step that is not used in chosen reference mode, this value is ignored.

- Referenced Position – after referencing was finished successfully, the current axis position is set to the position value specified here
- Limit Switch 1/2 Input – beside the soft limits it is possible to define hardware-controlled position limits via limit switches. When such limit switches are used and connected to E1701M controller, the inputs they are connected with can be specified here. Whenever one of these switches sends a HIGH signal, current movement is stopped using stop-deceleration
- Invert input logic – inverts the logic of the limit switch input to react on LOW level instead of HIGH
- Auto-leave Limit Switch – when a limit switch is hit, normally no more motions are allowed until this switch is left. When this option is set and an axis hits a limit switch, it automatically moves back until this switch is left.  
PLEASE NOTE: when this option is set, a possibly unexpected motion may start in opposite direction!
- Encoder – E1701M controller board supports up to two external quadrature encoders that provide position information. When such an encoder is connected to decoder input 0 (digital inputs 0 and 1) or decoder input 1 (digital inputs 2 and 3) these inputs can be enabled here for an axis. When an external encoder is activated newly, the axis position on start-up is set to 0. To specify a different position, reference mode "No referencing" has to be chosen, a referencing position has to be set and referencing has to be performed. This does not cause any referencing operation but immediately sets the current, encoder-based position to the one specified as "Referenced Position". Alternatively a standard referencing operation as described above can be performed too.  
PLEASE NOTE: an encoder can be used for one axis only, when the same decoder input is configured for more than one axis, the results are undefined.
- Encoder Factor – this factor is used for calculation of axis incremental position, here the ratio between encoder pulses and motor increments has to be specified

This plug-in is located in flowplugins/libio\_e1701m\_zshifter.

#### Sill(R) Focus Ethernet:

This is a one-axis motion plug-in to access a Sill focus shifter via Ethernet. Since the hardware is controlled via serial interface an Ethernet to serial converter is required in order to use this plug-in. In first configuration panel the TCP/IP configuration of that converter has to be set. In second panel the motion parameters can be set the focus shifter has to be operated with. Here a conversion factor from distance (in unit  $\mu\text{m}$ ) to internal representation of the focus shifter has to be given. Beside of that the upper and lower movement limit (in unit  $\mu\text{m}$ ) can be specified as well.

This plug-in is located in flowplugins/libio\_sill\_focus.

## 5.5.6 Process Controller Configuration

#### E1803D PID control loop:

This plug-in can be used together with the E1803D scanner controller card to access and use the integrated PID controllers of this scanner card. It can be used for controlling different parameters like temperature, pressure or concentration. The measured data have to be provided via the analogue inputs Aln0, Aln1 or Aln2 of this scanner card (this may require an appropriate extension board). These inputs directly correspond to the control loops PID0, PID1 and PID2 which can be configured separately:

- Enable PIDx/Alnx – enable the related PID controller for usage
- Type of value – here it can be chosen which type of data have to be measured; this does not have an influence on the behaviour of the PID control loop but is used to display proper names and measurement units within the main application
- Value at 0V – specify the temperature which belongs to an analogue input value of 0V
- Value at 5V – specify the temperature which belongs to the maximum possible analogue input value of 5V

- Lower value – the lower threshold for halting a building operation, when the measured value falls below of this, marking is halted until measured value is back in range “lower value” .. “upper value”; this value can be overwritten by parameters specified in pen settings
- Upper value – the upper threshold for halting a building operation, when the measured value rises beyond of this, marking is halted until measured value is back in range “lower value” .. “upper value”; this value can be overwritten by parameters specified in pen settings
- P – proportional part of the PID control loop
- I – integral part of the PID control loop
- D – differential part of the PID control loop
- Default setpoint – the default setpoint to be used on initialisation, this value can be overwritten by parameters specified in pen settings
- Control output – here the output can be chosen which has to be used to emit an output value depending on the current measured input value and the state of the control loop. It can be chosen between two analogue outputs AOut0 and AOut1 and the digital outputs from DOut0 to DOut7. Due to its limitations to an on/off state, the digital output should be used only for simple, slowly reacting environments, elsewhere analogue outputs are to be preferred
- Mode – this option can be used to configure the control mode and is only relevant when analogue outputs have been chosen. Here following options are available:  
*5V +/- regulation* – in this mode a 5V output value at the analogue output means setpoint and measured value are identically and no modification is necessary in process parameters. When measured value falls below of the setpoint, the output is regulated to values smaller than 5V (decay). When measured value rises above of the setpoint, the output is regulated to values bigger than 5V (amplification)  
*0V + regulation* – in this mode a 0V output value at the analogue output means setpoint and measured value are identically and no modification is necessary in process parameters. When measured value falls below of the setpoint, the output stays at 0V. When measured value rises above of the setpoint, the output is regulated to values bigger than 0V (amplification)

When this plug-in is used out of BeamConstruct, the pen parameters are extended by an additional tab-pane “PID ctrl loop” that offers some parameters that can be set to individual values for each pen. Depending on the number of PID controllers here up to three sets of parameters can appear. During marking these related values are sent to the plug-in whenever then pen changes modifying the current control loop and processing parameters:

- Lower value – the lower threshold for halting a building operation, when the measured value falls below of this, marking is halted until measured value is back in range “lower value” .. “upper value”; this value can be overwritten by parameters specified in pen settings
- Upper value – the upper threshold for halting a building operation, when the measured value rises beyond of this, marking is halted until measured value is back in range “lower value” .. “upper value”; this value can be overwritten by parameters specified in pen settings
- Default setpoint – the default setpoint to be used on initialisation, this value can be overwritten by parameters specified in pen settings

This plug-in is located in flowplugins/libio\_e1803\_pid.

Generic analogue temperature:

This plug-in can be used together with a scanner controller card which offers analogue input data in BeamConstruct environment. The behaviour of the plug-in can be configured. The first configuration-panel “Basic” provides following set-up possibilities:

- Analogue input port – here the port of the scanner controller card has to be selected where the temperature sensor is connected with (in BeamConstruct environment) or the type of analogue provided on binary IN7 the plug-in has to work with.
- Temperature at 0 – specify the temperature which belongs to an analogue input value of 0

- Temperature at max – specify the temperature which belongs to the maximum possible analogue input value

The second panel “Thresholds” gives the possibility to define the behaviour of the plug-in via following options:

- Lower Temperature – here a default temperature can be set which is used to turn on an external device (e.g. a heater), when measured temperature falls below of the limit specified here, the device is turned on; within a BeamConstruct environment this value can be overridden by pen parameters
- Upper Temperature – here a default temperature can be set which is used to turn off an external device (e.g. a heater), when measured temperature rises above the limit specified here, this device is turned off; within a BeamConstruct environment this value can be overridden by pen parameters
- Additional Digital Output Port – this parameter can be used within BeamConstruct only to configure the digital output port of a connected scanner controller card to be used for switching an external device
- Additional Digital Output Bit – this parameter can be used within BeamConstruct only to configure the pin number of the digital output port configured with previous parameter
- Stop Process on Temperature Deviation – when this option is set, a running marking process is stopped, when the temperature is above the upper temperature limit or below of the lower temperature limit by the given amount of “Temperature deviation”. The process is continued only when temperature is back within the given range (plus/minus the “Temperature deviation”).

When this plug-in is used within BeamConstruct, the pen parameters are extended by an additional tab-pane “Heating” that offers some parameters that can be set to individual values for each pen. During marking these related values are sent to the plug-in whenever then pen changes:

- Lower Temperature – a temperature which is used to turn on an external device (like a heater), when measured temperature falls below of the limit specified here, the device is turned on; this device is toggled via the output port and output bit specified in configuration panel “Thresholds”
- Upper Temperature – a temperature which is used to turn off an external device when measured temperature rises above the limit specified here; this device is toggled via the output port and output bit specified in configuration panel “Thresholds”

This plug-in is located in flowplugins/libio\_generic\_ana.

#### Optris LT Pyrometer:

This plug-in can be used together with an Optris LT pyrometer with serial interface to read temperature data from it and to react on the measured values accordingly. The behaviour of the plug-in can be configured. The first configuration-panel “Basic” provides possibilities to set-up the serial interface, here the same parameters have to be specified the Optris LT serial port is running with. Also to be configured in the first panel is the protocol format to be used when communicating with the pyrometer, here the option to be selected depends on the exact device which is used.

The second panel “Thresholds” gives the possibility to define the behaviour of the plug-in via following options:

- Lower Temperature – here a default temperature can be set which is used to turn on an external device (e.g. a heater), when measured temperature falls below of the limit specified here, the device is turned on; within a BeamConstruct environment this value can be overridden by pen parameters
- Upper Temperature – here a default temperature can be set which is used to turn off an external device (e.g. a heater), when measured temperature rises above the limit specified here, this device is turned off; within a BeamConstruct environment this value can be overridden by pen parameters
- Additional Digital Output Port – this parameter can be used within BeamConstruct only to configure the digital output port of a connected scanner controller card to be used for switching an external device
- Additional Digital Output Bit – this parameter can be used within BeamConstruct only to configure the pin number of the digital output port configured with previous parameter

- Stop Process on Temperature Deviation – when this option is set, a running marking process is stopped, when the temperature is above the upper temperature limit or below of the lower temperature limit by the given amount of “Temperature deviation”. The process is continued only when temperature is back within the given range (plus/minus the “Temperature deviation”).

When this plug-in is used within BeamConstruct, the pen parameters are extended by an additional tab-pane “Heating” that offers some parameters that can be set to individual values for each pen. During marking these related values are sent to the plug-in whenever then pen changes:

- Lower Temperature – a temperature which is used to turn on an external device (like a heater), when measured temperature falls below of the limit specified here, the device is turned on; this device is toggled via the output port and output bit specified in configuration panel “Thresholds”
- Upper Temperature – a temperature which is used to turn off an external device when measured temperature rises above the limit specified here; this device is toggled via the output port and output bit specified in configuration panel “Thresholds”

This plug-in is located in flowplugins/libio\_optris\_lt.

## 5.6 Machine Configuration

There is a set of machine-dependent parameters which are not stored within each project file or in default project configuration, but locally on each machine where BeamConstruct is installed at. These parameters can be configured via related dialogue in menu “Project” sub-menu “Machine settings...” which offers several configuration options within different tab panes. The following parameters can be set and modified within these panes:

In tab-pane “User interface”:

- Expand / Move entity tree to tab panel – this option saves some space in X-direction; the entity tree that is shown on main windows right hand side normally will be moved to the left side and becomes a normal tabbed panel there; changes to this parameter do require restarting of the software in order to take effect
- Security / No warning message at start-up – in general it is not recommended to use this option, it makes sense in some very rare cases only: when this checkbox is set, the warning message about the possible dangers of this application is no longer shown. This means, prior to turning of this warning, other ways of informing the user have to be established in order to avoid unwanted and possible dangerous usage of this option.  
Changes to this parameter do not require saving of the current configuration as default one, the related information is stored automatically and always only locally, it does not belong to a project file which can be given away.
- No license progress bar on start-up – on application start BeamConstruct checks for a valid license. Depending on the speed of the used computer this can take some seconds. During this time a progress bar is shown. When this option is set, this progress bar is no longer used but license checking is still done – so with this option the delay at application start up is still there but without a visual feedback.
- Switch back to old OpenGL 1.1 – the 3D view within the application makes use of modern OpenGL for fast drawing also of complex geometries and 3D models. On some graphics hardware and in case of poor quality of drivers of such graphics card this can cause issues – starting from wrong or corrupt displaying of models and ending up in a crash of the software as soon as the 3D view is opened. In such cases it is recommended to turn on this option to switch back to the slower OpenGL version 1.1
- Use preview-projector – normally the pilot laser is used to generate a preview of the current project to be marked. As this method is limited in terms of accuracy and visibility of details, BeamConstruct offers the possibility to use a digital or video projector to show the current projects data directly within the working area. For details about set-up and usage of such a projector for immersive production processes please refer to related section “5.6.1 Preview Projector” below.
- Hide support structure settings – when this option is set, the parameters to generate support-structures in 3D Slice Group tab pane is hidden from the user and no support structures can be generated any more
- Disable centre point – every element that contains some geometry comes with a centre point that is used for rotations of this geometry. This centre point can be edited, means it can be moved to any suitable position relative to the geometry itself. When this option is enabled, this centre point is no longer visible when an element is selected and it no longer can be repositioned.
- Disable 3D view – beside the flat, 2D views, there is also the possibility to switch to a 3D view mode. For applications where this 3D view does not make any sense, this option can be used. It disables the 3D view completely, it will be no longer available, also in case the scanner controller card is configured to support the third axis
- Language – by default this parameter is set to “Automatic”, means the application chooses the language the underlying operating system is configured for. When other translations are available, they are listed in the related combobox and can be chosen to be used instead. Then on next start of the application, this language is used for the user interface.
- Menu item shortcuts – the area below of these parameters contains an editor that gives the possibility to edit the shortcuts of BeamConstructs menu items. On lower left hand side the menu item to be edited can be selected in a tree view. Next right beside that tree the new shortcut has to

be typed in field “New Shortcut” and then can be added using button “Add”. The new shortcut is shown in the list right beside the menu tree view and can be removed from there using button “Remove”. After the settings dialogue is left with “OK” these new menu shortcuts are applied to the menus.

These settings are locally only and do not become part of a project.

The tab-pane “Toolbars” gives the possibility to configure which tools are shown in the toolbars and which are hidden in order to not to be accessible by the user. With the selection list at the top the toolbar to be configured can be chosen. Next the list at the right shows the tools that are accessible. Selecting such an entry gives the possibility to move it to the left (the list of hidden and therefore not accessible tools) by pressing the button with the arrow to the left. In same way it is possible to chose a hidden tool and move it to the right list by pressing the button with the arrow to the right.

After changing the tools within the toolbars, the software needs to be restarted.

Please note: tools, which already have been disabled by the related customisation functions (as described in “5.19.3 Custom menus and functions”), are not visible here, neither in the “shown” nor in the “hidden” section. Furthermore, in case of custom installations, where specific tools have to be disabled on set-up of the software, it is recommended to not to use the toolbars-function out of the machine settings, but to use the customisation functions out of the section 5.19.3 too.

The “Smart Factory” tab-pane offers functionalities which are related to machine integration into automated production lines:

- Machine name – here a name can be set which identifies a software installation/a machine uniquely. By default an auto-generated name is set here which can be replaced by a human-readable value. This name is used for all Smart Factory related functions, e.g. within all Smart Interface state messages to identify the machine it comes from and within the relevant Hermes messages that require a unique identification too.
- Enable Smart Interface – when this checkbox is set, the Smart Interface is enabled and the software can be integrated into a Smart Factory / Industry 4.0 environment. For details regarding this interface please refer below
- Enable MQTT connection – when this checkbox is set, state messages are sent to a suitable broker using the MQTT protocol
- Broker – here IP and port number of the broker to publish the MQTT state messages to has to be set
- Topic base-name – using this field a topic for the MQTT messages can be configured. It will consist of a base-part of the topic which always is expanded by the machine name configured above
- Enable Hermes automatic control – when this checkbox is set, support for machine integration according to the Hermes-standard is enabled (for details please refer to <https://www.the-hermes-standard.info>). This enables several configuration parameters which can be used to specify connection to other machines within the same line. Here it depends on the availability of previous and/or next machine if BeamConstruct acts as end of line machine, begin of line machine or something in between. If the next or previous machine does not connect according to the Hermes specification, BeamConstruct automatically switches to end/begin of line mode.
- Previous machine – here IP and port number of the machine which transfers working pieces to BeamConstruct can be configured
- Next machine – here the port number of the machine BeamConstruct transfers working pieces to can be configured. Since the next machine connects with BeamConstruct actively, here no IP has to be specified. Instead of this the local IP must be known at the next machine.
- Conveyor axis – using the following comboboxes it can be specified which motion controller and which of its axes has to be used as conveyor to feed the working pieces. Here also the movement direction of the conveyor can be set.
- TransportFinished distance – when the conveyor axis moves to transport a working piece to the next machine, a TransportFinished command has be sent to the next one, as soon as this transport is complete. Using this field the motion distance can be specified which has to be used to move a working piece out of the current machine. This parameter is used only when BeamConstruct does not act as end-of-line machine.



- Enable Hermes remote configuration – according to the Hermes standard there is a possibility to send configuration data from a remote host and to overwrite local information as they may have been specified here. When this checkbox is set, overwriting of the local configuration is allowed and possible. When it is not set, the Hermes-configuration of this machine can be changed only by direct manual interaction.

In tab-pane “Users” access to functions can be limited depending on logged-in users, their roles and privileges. By checking the box “Enable user management”, this functionality is activated and some predefined settings are provided. For all existing, preconfigured users the password is similar to the username and should be changed prior to operating it on the target system.

The users-tab-pane lists the available users on the left hand side. By selecting one of them, the related privileges and settings are shown on the right hand side of the pane. Once a user is selected, its privileges can be changed, the login and full name as well as the state and the comment can be modified. To activate such changes after setting them, the button “Apply” has to be pressed, elsewhere the changes of a users parameters are not activated.

The check-boxes on upper right side of this tab-pane define what a user can do after logging into the software successfully:

- Supervision – this is a special privilege that applies only to user “Supervisor”, as long as this user is logged in, all functionalities are enabled, no restrictions exist
- Manager Users – a user with this privilege is allowed to modify the users and its privileges within this tab-pane; this should be combined with the “Modify Settings” privilege in order to give the related user access to the configuration dialogue
- Exit Application – this privilege gives a user the possibility to leave the application
- Modify Settings – a user with this privilege is able to enter the project configuration dialogue and to change the global parameters and settings of an installed software
- Edit Geometry – when this privilege is set for a user, this user will be able to modify loaded geometry and to edit the data of the currently loaded project
- Load Project – this gives a user the possibility to load an other project, this privilege is similar to “load recipe” like it may be named in other software
- Save Project – a user with this privilege is allowed to save the current project to disk; this privilege should be combined with an other privilege that gives the possibility to change the project settings, to edit the loaded project or to load/insert an other project, elsewhere it would not make sense for this user to save an unchanged project only
- Control Marking – when this privilege is set, a user is allowed to start and stop the marking process

These user data are stored in the default application data directory of the used operating system within a sub-folder “beamhome”. The exact position depends on the used operating system. The data are neither encrypted nor protected, so deletion of the directory and its contents would disable the user management completely. Thus it has to be ensured by using the operating systems functionalities that no user is able to access this folder without permission (e.g. by disallowing users to exit the application and by locking the whole system as long as BeamConstruct is running). This is done by intention to offer the possibility to restore a system just by deleting the “beamhome” directory when some relevant passwords got lost.

In case the user management was activated, on next software start-up all functions are disabled until a valid user logs in by using the correct password. Logging into the software can be done by pressing the user-



### 5.6.1 Preview Projector

To get a more detailed and more complete preview of the current project to be marked, as an alternative to the slow and limited pilot laser a projector can be used that is adjusted to display its contents directly into the marking area. BeamConstruct then offers extensive possibilities to use and control such an additional projector.

Following preconditions have to be met in order to make use of such a comfortable, additional view to the project to be marked:

- a PC with a second video-output is needed that can be configured to display separate contents on a screen
- a projector (video or digital projector) is needed which can be connected to this PC in order to display the contents visible on this second screen

To install and adjust such a projection system, following steps are necessary:

- mount the video projector in your machine at a position where it can cover the working area completely
- configure the PC in a way so that each of the available video outputs (aka “screens”) shows separate contents and an own desktop (“Extend displays”)
- reposition BeamConstruct’s main window so that it is shown on the first screen (the screen with the system-toolbar)
- connect the projector (aka “Beamer”) to the video output that is assigned to the second screen
- go to menu “Project” → “Machine settings” and enable the function for the preview projector
- leave the machine settings dialogue with “OK”; now a new window appears which contains the preview which has to be shown by the projector
- move the new window to the second screen and maximize it so that it fills this second screen completely – to simplify this at this stage the frame around this window is still visible
- create a project in BeamConstruct that contains some geometries which can be used to adjust the projected view in next steps; this typically is a rectangle which is as large as the whole working area as well as some additional graphics which mark specific positions within the machine (e.g. a cross-hair to mark the centre position); this projects vector data now are also shown via the beamer and projected into the machine, but still not at the correct position and spatially distorted
- go to menu “Help” → “Adjust projector”; now a dialogue opens which offers different functions to correct the view of the projector in order to be perfectly aligned to the working area also in case the beamer is mounted angular and at the side of the machine which causes some spatial distortions
- first uncheck the box “Enable window dragging border” to turn off the frame around the projection window within your projectors view; now it is impossible to move the projection window any more
- scale and adjust the projected view in the machines so that it fits to the real world working area using the available adjustment parameters:
  - Offset X / Offset Y – to move the view in X and Y direction
  - Zoom – scale the whole view
  - Spatial correction X / Spatial correction Y – adjust the spatial distortions caused by the non-parallel mounting position of the projector
  - Rotation – rotate the whole view
 these values can be changed either in larger steps by pressing the related “+” or “-” buttons or by entering exact values into the related input fields. When the view got lost accidentally and can’t be brought back to the projection any more, enter a value of 1 to the zoom-input field and a value of 0 to all other input fields
- leave the dialogue with “OK” in order to let the current settings be saved

Now whenever a project is loaded, the contents of this project are shown within the working area too. Here a different colouring scheme is used for a better contrast and brightness within the machine, so the colours for pens background which have been configured in pen and project settings are not used here.

## 5.7 Pen Settings

Every element that is created within a BeamConstruct project has a pen assigned to its geometry (or for scanner bitmaps to its bitmap data). This pen contains all information the related element will be marked with: scanner speeds, delays, laser power and frequency and others more. These parameters can be set within the pen settings dialogue that opens when the menu “Project” menu item “Pen settings...” is selected.

On top of that dialogue the head can be selected this pen applies to. This is important for multihead environments where more than one scanner controller/more than one independent working area is used. There will always exist the same number of pens for all heads but the same pen number can have different parameters for different heads.

Below of this a pen can be chosen for editing, the same combo box can be used to create a new pen, in this case the option "Append new pen..." has to be selected there. After selecting an existing or a new pen all its parameters can be changed within the lower part of the dialogue. It is separated into some sections that contain different categories of pen parameters.

All tab-panes that hold these categories have one common functionality for selected parameters: by pressing the button "To all pens" a parameters value can be applied to all pens that currently exists. So in case a specific pen parameter has to be changed globally, with this function it is not necessary to edit it manually within every single pen.

PLEASE NOTE: there is no feedback from a possibly connected laser or scanner equipment to this dialogue, so the application can't know if a value is suitable for the connected equipment or not. Although there are some logical and range checks that try to avoid heavy mistakes it is up to the user to keep all parameters within valid and useful ranges.

### 5.7.1 General Pen Settings

This panel contains some basic pen parameters. On top of it an input field can be found that can be used to rename the current pen, there its name can be changed and the new name can be applied by pressing the return key.

Beside of that the following parameters can be edited here:

- Power – the output power in range 0..100% the laser has to be driven with
- Frequency – the frequency in unit Hz the laser has to be driven with; the allowed frequency range can be configured within project configuration dialogue
- Mark Speed – the speed of the scanner that is used during marking (motion while laser is turned on); the allowed speed range can be configured within project configuration dialogue
- Jump Speed – the scanners speed that is used during jumps (motion while laser is turned off); the allowed speed range can be configured within project configuration dialogue
- Spot Size – the diameter of the lasers spot on the surface, this value should be as exact as possible because it is used for different calculations
- Draw Off-Movements – this option is related to the visual representation of a project within BeamConstruct only; when this option is set not only these parts of the geometry are shown within the drawing area where the laser is turned on but also the jumps where the laser is turned off; this option can be helpful to optimise a project and to re-arrange elements to perform as less jumps as possible
- Use as default Pen – the pen where this option is set for is used as default pen for all newly created geometries. More than this, whenever a new pen is created the parameters of this pen will be set to the values specified for the default pen. Within a project only one pen can be default, so whenever this option is selected for a pen, it is disabled, resetting this option can be done only by choosing an other pen as default one.
- Use for Pilot-Laser – the parameters of this pen are used to control the pilot-laser. This means the speed and delay settings are used whenever the pilot laser is started. Since this laser can be turned on and off only, all values like power, frequency and some delays are disabled, they are not relevant for the pilot laser. Within a project only one pen can be used for the pilot laser, so whenever this option is selected for a pen, it is disabled. Resetting this option can be done only by choosing an other pen for the pilot laser.
- Pen On-Colour – this colour is used within the drawing area for all geometries
- Pen Off-Colour – this colour is used when "Draw Off-Movements" are enabled to show the laser jumps within the drawing area

- Use for coloured scanner-bitmaps – when this checkbox is set the current pen is used for coloured scanner bitmaps; this option should be enabled only
  - 1) when the current pen parameters cause a coloured result on current material
  - 2) when the colour of this result is specified using button “Colour after marking”; PLEASE NOTE: setting the resulting colour requires a calibrated monitor in order to have exact results

### 5.7.1.1 *Pen Parameter Wizard*

It is sometimes difficult to find parameter-combinations that fit to a special material or application. Thus the pen dialogue offers a special wizard that creates a matrix out of different values where the user can choose a suitable result from. To open this wizard, the button “Parameter Wizard” on upper left corner of the pen settings windows has to be pressed.

There a window opens where a 4x4 to 10x10 sized matrix of small marking areas is symbolised by a matrix of buttons. On upper part of the window the parameters to be used have to be selected and their range to be checked can be entered. These values are spread over the matrix in order to find best parameters. Next the yellow “Mark Start” button right beside the button matrix has to be pressed in order to start output to the material. Alternatively it is possible to use the button “As project” in order to not to mark out of the pen parameter wizard directly but to use the generated pen parameter matrix as current project which then is editable like every normal BeamConstruct marking project.

ATTENTION: by pressing the mark-button a laser operation is started! This may operate with up to 100% power and therefore is potentially dangerous! Prior to opening the pen parameter wizard and prior to starting output out of this wizard all security measures have to be ensured!

As soon as marking has finished, the area with the best marking result has to be chosen from marked material and the related button in pen parameter wizard has to be pressed.

To make it easier to find the correct position there is one circle and one triangle used for marking, these symbols identify orientation and direction of the matrix uniquely. So to find the position of the best result it is possible to first count from triangle to circle until the correct column is found and then count the row number. When the same is done in pen parameters button matrix of the wizard dialogue, the exact position is found and this button can be pressed. Such a button-press opens a popup-menu where one of the following options can be chosen for further processing:

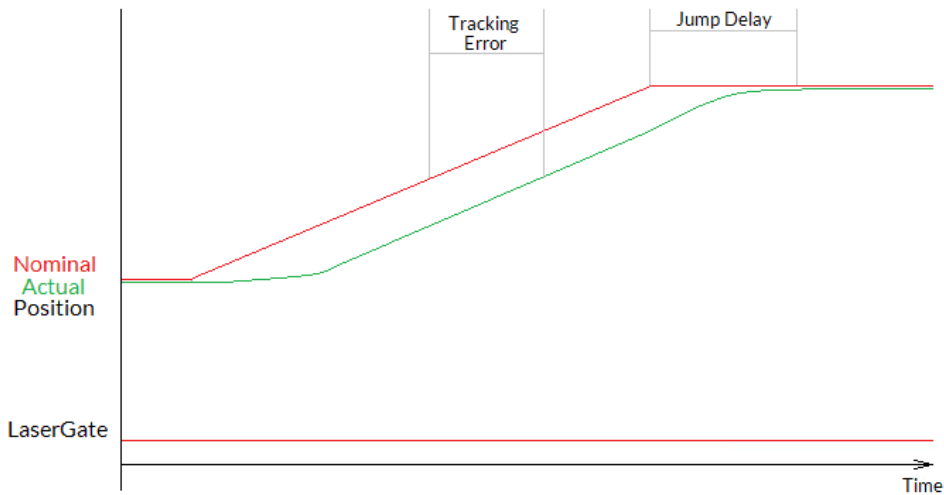
- Refine this result – some values close to the chosen position are used to create a new matrix for a next, more accurate calibration step
- Go back from this result – this option performs the opposite operation of the previous one, it creates a new parameter matrix with a larger range of values and should be used when a parameter range out of a refined result or when the initial parameter range does not give expected marking results
- Use parameters for pen – when this item is chosen the parameter wizard will be closed and the found parameters that belong to the selected field are set to the current pen.

For the other marking parameters apart the selected ones the pen parameter wizard always used values of the currently selected pen, only the selected ones are varied and modified by this function.

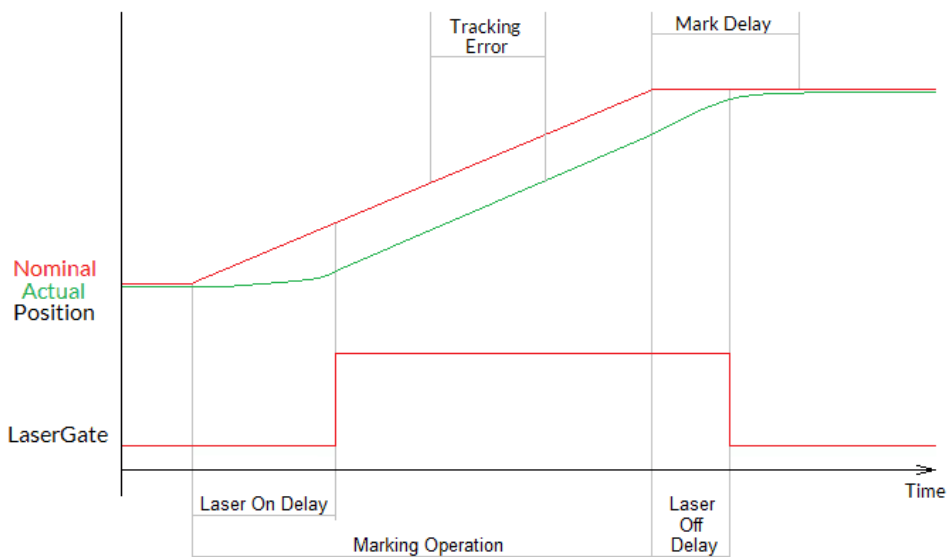
### 5.7.2 **Scanner Pen Settings**

This tab-pane contains all scanner-related pen parameters:

- Jump Delay – time the scanner waits when a jump was executed; this delay applies at the end of a jump to let the scanner finish its movement



- Mark Delay – time the scanner waits when a marking operation was executed; this delay applies at the end of a marked line to let the scanner finish its movement



- Polygon Delay – the time the scanner has to wait for every corner within a polygon when the laser is turned on and stays turned on
- Use variable polygon delay – this option belongs to parameter Polygon Delay. When it is set, there is no static delay used in every corner of a polygon, but a variable one which depends on the current angle between two lines of the polygon. The sharper the angle is, the longer the used polygon delay becomes. This option is available only when a scanner controller card is configured which supports this feature.
- Auto-generate delays – the delay values described above are essential to get high quality marking results also in case of an as high as possible marking speed. Since the evaluation of the correct values is a difficult and time-consuming process, BeamConstruct offers a possibility to set some default values which can be used as starting point for optimising these delays. After pressing this button, a dialogue opens, where the used scan head can be selected out of a list of predefined types. Then some base delay values are generated for this scan head. In case a used scanhead is not listed there, option “Other” has to be chosen, which requires to enter one single value for delay calculation: Every scan head vendor should provide a value named “time lag”, “tracking error” (or something similar) within its technical documentation. This value describes the specific inertia of this scan head and needs to be entered here. When this dialogue is left by pressing “OK”, BeamConstruct calculates some basic delay parameters (including not only the jump- and mark delay but also laser-on and -off delay as per section “5.7.4 Laser Pen Parameters” below) that fit to the scan heads mechanical properties and often can be

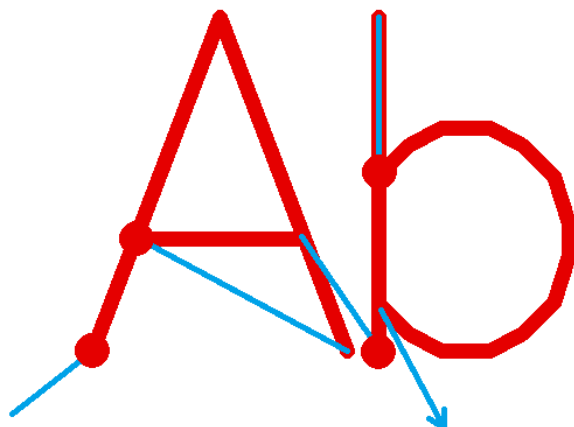
used directly. Alternatively these delays can be used for further optimisations for these applications, where a more strong and exact set-up of the scan head is required.

For more details about the purpose of all these delays and their influence please refer to your scan heads or scanner controller manufacturers manual.

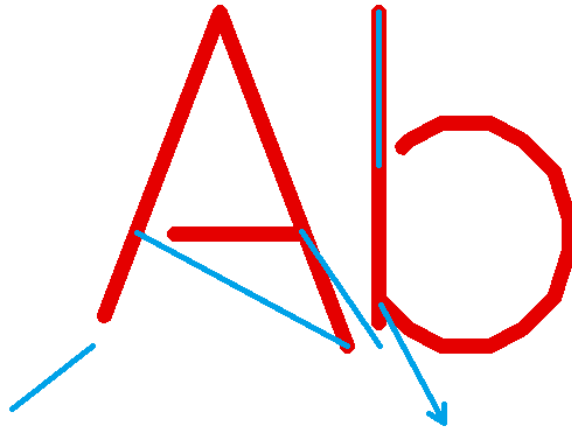
- Set Defaults for Bitmap-Marking – when marking scanner bitmaps instead of vector data, some very specific pen parameters have to be used that follow other rules. In such cases it is recommended to use this button, it changes the current pen parameters including the scanner delays to values that are suitable for marking scanner bitmap elements.
- Wobble mode – when wobble is enabled the linear movement of the scanner during marking is distorted by an additional frequency that lets the scanner oscillate around the predefined position. This functionality can be used e.g. to warm up larger areas of a material, for welding or similar things. To enable wobble, a mode has to be selected and both, a wobble frequency and amplitude have to be set.  
Following wobble-modes are supported:  
Straight: wobbling is done with fixed values for frequency and amplitude resulting in a regular wobble shape, in this mode the range-parameters as well as the update frequency parameter are ignored  
Fuzzy: the wobble parameters are randomly modified which results in a fuzzy-looking wobble-pattern; for this mode at least one of the range-parameters as well as the update-frequency value have to be set  
Sweep: the wobble parameters are changed linearly varying from one extreme (as configured by the “+” parameters) to the other
- Wobble Frequency / Wobble Amplitude – when both of these values are set to 0, the wobble mode is turned off, independent from what mode is configured. To do some wobbling, both parameters amplitude and frequency have to be greater than 0. The wobble frequency specifies how fast to oscillate and the wobble amplitude how much to vary over the original position. Wobble has to be supported by the used scanner controller card in order to have any effect during marking.
- +- Wobble Frequency / +- Wobble Amplitude – these parameters apply to the wobble frequency and amplitude when a non-straight wobble mode is configured. They specify a range, which defines, how strong the related wobble parameters have to be changed. When both are set to 0, this is similar to wobble-mode “straight”.
- Wobble update – when a non-straight wobble mode is set, this parameter specifies, how fast the wobble-parameters have to be changed

The given delay values have a direct influence on the marking result and need to be adjusted carefully. Wrong delay values can result in inaccurate geometries, burn-in points at some specific positions and/or unnecessary long marking times. Out of the visible result it can be evaluated which delays are too small or too big by checking following rules:

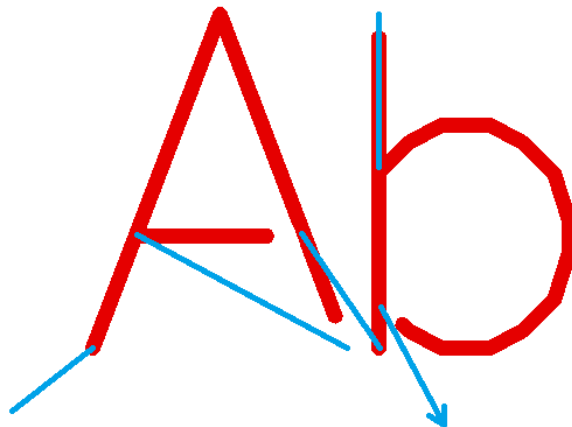
Parameter	Value too big	Value too small	Remark
Laser On Delay	First part of a mark vector (following a jump vector) is not marked	Burn-in point at the beginning of a mark vector that follows a jump vector	See pictures below
Laser Off Delay	Burn-in point at the end of a mark vector and before the next jump vector	Last part of a mark vector (before a jump vector) is not marked	See pictures below
Jump Delay	Mark time is longer than necessary	Geometry is damaged at the end of jump (transient oscillation)	
Mark Delay	Mark time is longer than necessary	The end of a vector is turned towards the direction of the following jump vector	See picture below
In-Polygon Delay	Burn-in points at the corners	The corners of the polygon are not sharp but rounded and at wrong position	



**Laser-on delay is too short**, laser is switched on too early while mirrors are still slow or accelerating, thus it fires on the same position for a too long time which results in burn-in points; please refer to section “5.7.4 Laser Pen Parameters” below for this parameter

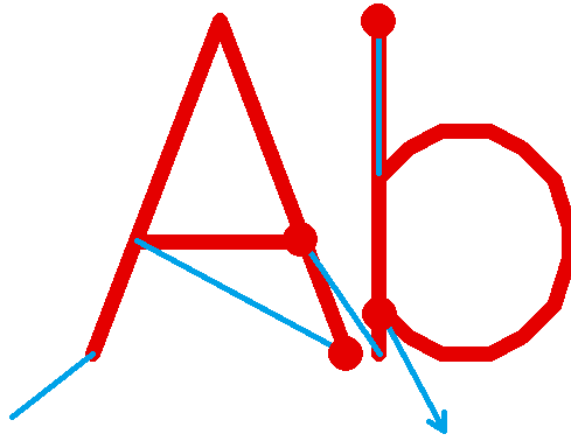


**Laser-on delay is too long**, laser is turned on too late while scanhead already started marking of the next vector, thus parts of the mark geometry are missing at the beginning; please refer to section “5.7.4 Laser Pen Parameters” below for this parameter

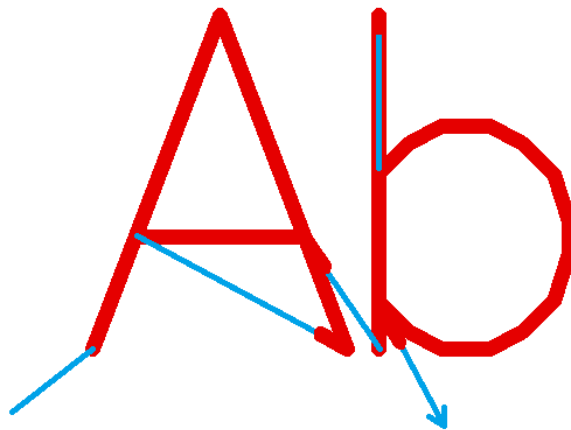


**Laser-off delay is too short**, laser is turned off before the end of the current mark vector is reached, thus parts of the mark geometry are missing at the end; please refer to section “5.7.4 Laser Pen Parameters” below for this parameter

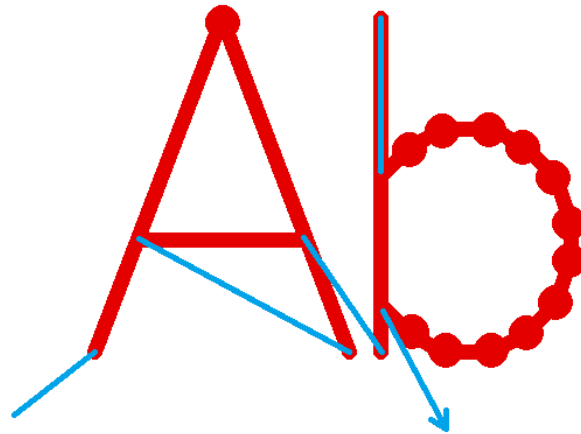




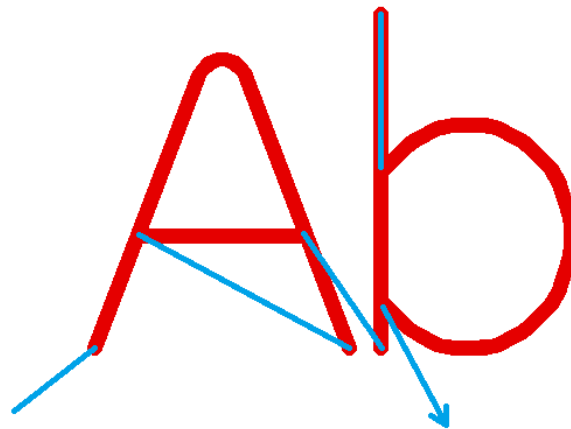
**Laser-off delay is too long**, laser is turned off too late while end point of the vector is already reached and mirrors of the scanhead slow down, this results in burn-in points at the end of the mark vectors; please refer to section "5.7.4 Laser Pen Parameters" below for this parameter



**Mark delay too short**, motion of next jump vector is started too early and before end position of previous mark vector was reached, thus the laser is still on when motion for the following jump vector is performed



**In-polygon delay too long**, scanner stops for a too long time at the position of a corner point within a polygon, thus burn-in points appear at these positions.



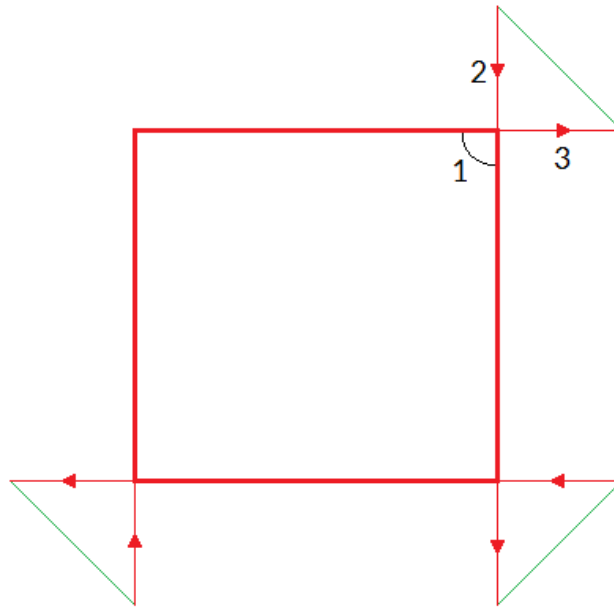
**In-polygon delay too short**, mirrors are not able to arrive at the positions of the corner points within a polygon before the next mark vector starts, thus the whole geometry is rounded, the vectors are pulled into direction of the next one

### 5.7.2.1 Sky-Writing

As described in previous section, it is necessary to apply delays in order to have clear and sharp corners within polygons. Depending on the used hardware and its capabilities these delays can become quite long, slowing down the total marking speed. To avoid this, the “sky-writing” mode is available (on selected scanner controller hardware which supports this feature or has required capabilities). When this mode is active, the configured polygon delays are no longer used which waste some marking time. On the other hand the vectors to be marked become longer which itself again needs some time. So it depends on the used hardware and its specific capabilities if sky-writing really saves time during marking.

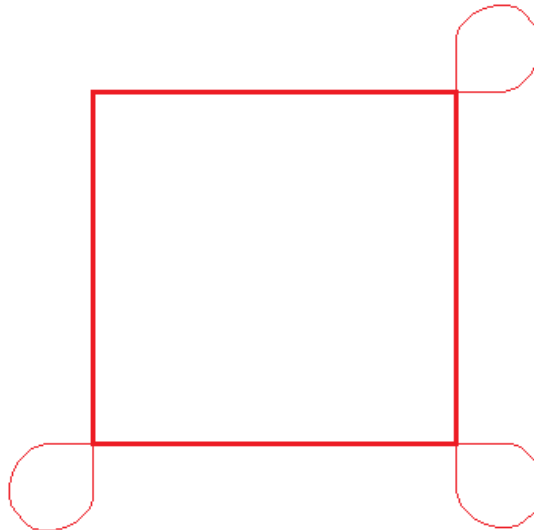
Sky-writing avoids delays and rounded, blurred corners by starting vectors earlier, ending them later and turn on/off the laser at exactly the position where the polygons line begins/ends:

In this picture the thick red lines symbolise the marks where laser is turned on. The thin red lines show the movements where vector lines are extended by a given length but where laser is already turned off. For a better understanding small arrows in the jump lines show the marking direction. The thin green lines show



the jumps, only at these positions the chosen jump speed is used, all other movements are done with mark speed.

Here each line of the rectangle starts earlier, turns on the laser at the point where the polygon line has to begin, turns it off at the point where it has to end and then continues a bit in the same direction. Next a jump is done to the position where the next extended vector line has to start. Now when the scanhead would distort vectors because there is no more in-polygon delay active, this distortion would happen outside the rectangle and at a position where the laser is turned off. Thus the distortion would no longer be visible in the marking result when the lengths given for the polygon are long enough:



In this image the real scanner movements are shown. Here the movements at the corners are rounded and distorted due to the high speed, but these distortions do not influence the marking result since they happen when the laser is already turned off. In this specific example the lengths the lines are extended by could be shortened because there is still a longer straight line before and after every line.

Sky-writing requires following parameters to be configured:

- activation threshold – marked with a “1” in picture above: this is a value that specifies at which angle between two lines within a polygon the sky-writing mode needs to be activated, if an angle is equal or smaller (means sharper) than this value, sky-writing is used to keep the marking result clear, for bigger (means flatter) angles the normal in-polygon delay (as described above) is used
- fade-in distance – marked with a “2” in picture above: this value specifies the length a line has to be extended with at the beginning, means this is the size which is added before laser is turned on. This value should be as small as possible in order to not to waste marking time. At the borders of the

current working area it may be possible this value can't be applied fully because the starting point would be located outside of the valid range of the scanhead.

- fade-out distance – marked with a “3” in picture above: this value specifies the length a line has to be extended with at the end, means this is the size which is added after the laser was turned off. This value should be as small as possible in order to not to waste marking time. At the borders of the current working area it may be possible this value can't be applied fully because the end point would be located outside of the valid range of the scanhead.

### 5.7.2.2 Overspeed

An other method of getting a higher speed out of a scanhead is the overspeed-functionality. With this feature the galvos of a scanhead are driven with a speed that is far beyond of its limits and normally results in a totally distorted marking result. Here the scanner controller card takes care about the movement of the galvo and drives it with a specific speed curve that avoids these distortions. When this feature is used, the exact improvement highly depends on the used marking data. The more long vectors it contains and the less inertly a scanhead is, the higher is the possibility to find vectors that can be driven with overspeed and the better is the possible improvement in speed.

To make use of the overspeed-function, some specific requirements needs to be met:

- the used scanner controller card supports these speed curves for overspeed movements
- scanhead bases on galvo-technology
- scanhead does not artificially limit the maximum speed

The overspeed-function can be enabled and configured in pen settings, tab-pane “Scanner”. Here following parameters can be set:

- Enable overspeed – enables the overspeed-function for the related pen; when this checkbox is set, the following parameters need to be configured:
- Tracking error – this is a scanner-specific value (also named “time lag” sometimes) which describes the inertia of the mechanical parts of the used scanhead. This value is fixed for each type of scanhead and is typically provided by the vendor of the scanhead.
- Jump speed factor – this value is the overspeed factor which is used for jumps. Normally jumps are done using the speed defined in tab-pane “General”. When the overspeed-feature is enabled and when a jump vector is suitable to be driven faster, it no longer is moved with the nominal jump speed but with the nominal jump speed multiplied by the factor given here.  
The maximum value this factor can have depends on the used scanhead but typically is in range 3..9

To adjust a scanhead for usage with the overspeed-function and to find suitable parameters, please perform the following steps:

1. Create a test-project that represents a typical application but also contains some large jumps (at least half of the working area is recommended). These large jumps later are used to find the maximum overspeed factor
2. Start with the overspeed-function disabled.
3. Use the test project from step 1 to find the proper scanner and laser-delays as well as the nominal maximum jump and mark speed the scanner can be driven with while it still provides acceptable marking results with no distortions (this is the normal procedure of setting scanner parameters, so please refer to previous sections about adjustment of these delays).
4. Enable the overspeed-function for the pen(s) used in the test project and set the tracking error value according to the technical data of the used scanhead.
5. Increase the jump speed factor until you find a value where the marking results start to become distorted, here especially the marking results which are located close to the long jump vector(s) are important and have to be checked

### 5.7.3 Bitmap Pen Settings

Within the panel “Bitmap” some scanner-bitmap-specific parameters can be set. They are used only, when this pen is assigned to a bitmap but has no effect when it is used with vector data.

Please also note the button “Set Defaults for Bitmap-Marking” which is described in section “5.7.2 Scanner Pen Settings” above.

This panel provides the following parameters:

- **Bitmap greyscale mapping** – in some cases a linear 0..100% power range does not result in similar, linear grey tones. This happens when the laser does not react linear on provided power values and/or when the material does show a non-linear behaviour on marking. In such cases this option can be enabled to define an own greyscale mapping which allows to adjust such deviations. On upper part of the window a nominal (linear) greyscale range is shown. Below of these the mapping values can be entered which have to be assigned to the nominal, input grey values of the bitmap to be marked. Below of these sliders and input fields the resulting greyscale map is shown as it would be sent to the laser in terms of power values. When the given values are correct, this non-linear, lower greymap then would result in linear marking results. The given values are in range 0..255 which corresponds to the 8 bit greyscale values of a scanner bitmap. This greyscale map is used only with greyscale bitmaps where a pen is assigned to in which this option is enabled. To adjust a bitmaps greyscales with this map, the test image from [https://halaser.eu/scannerbitmap\\_test.png](https://halaser.eu/scannerbitmap_test.png) can be used – beside different patters which can be used to evaluate the quality of a greyscale marking it contains seven grey values in lower right corner which directly correspond to the seven grey parameters which can be set with this option. So as soon as these seven fields have the correct brightness after marking, the given greyscale map is correct.

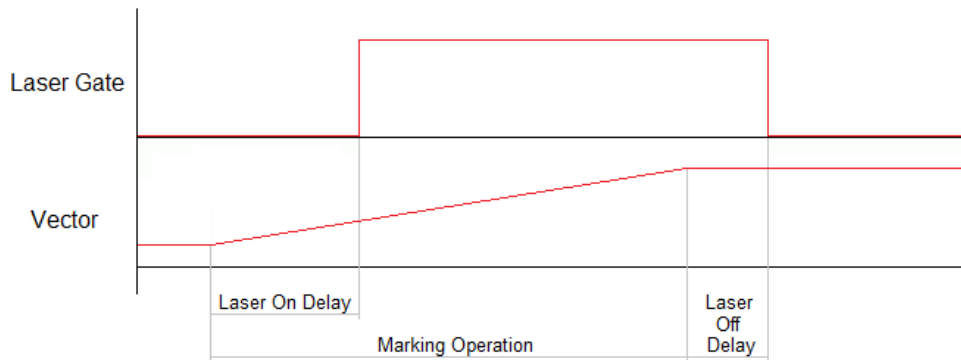
### 5.7.4 Laser Pen Parameters

This panel contains several parameters that belong to the used laser and the parameters/signals that are required to operate it. So depending on the laser type that was configured for the current project these options are enabled that contain useful parameters for proper laser operation. For a detailed description of these parameters and their meaning please refer to your laser vendors manual.

PLEASE NOTE: some laser types need to be operated using relatively tight parameter ranges. When they are controlled with wrong parameters the results may be an undefined behaviour or the laser hardware may be damaged in worst case. So when setting up new parameters or when changing them or after a software update was done, it is highly recommended to (re-)check these laser parameters and the output signals of the scanner controller in order to avoid problems with the laser. Here “software update” not only includes updates of the HAL software package or BeamConstruct. Also in case the software or drivers that for an example belong to the scanner controller card are modified, it is recommended to perform these checks.

This tab-pane contains all laser-related pen parameters:

- **Laser On Delay** – the time that is set between start of a mark movement and the moment the laser is switched on; so this parameter specifies if and if yes how much the laser should be turned on delayed in case a marking operation is started
- **Laser Off Delay** – the time that is set between the end of a mark operation and the moment the laser is turned off; so this parameter specifies how much turning the laser off has to be delayed in case a marking operation ends



- Dot-length – this is a convenience function which shows and calculates the number of pulses a dot consists of with the current settings. The value given here depends on the frequency and the current laser on and laser off delays.  
So when the value in this field is changed, the corresponding laser off delay is changed automatically. On the other hand, when frequency, laser on delay or laser off delay are changed, the number of pulses shown in this field are changed according to the modified values.

### 5.7.5 Using Pens for Coloured Bitmap Marking

On some materials variations of power, speed and frequency cause marking results in different colours. This behaviour can be used within BeamConstruct to easily mark coloured bitmaps. Following the steps are described to get such images. First of all it is important to know that the process of marking colours is not easy to handle. Environmental conditions may have a strong influence the result so that pen parameters that cause a specific colour one day may result in a different colour the next day. So the whole process should be done in a well-defined environment. Next the material condition can have an influence too, its state of oxidation, grease or dirt on the surface may affect the resulting colour. So in case of troubles a defined cleaning process may be necessary, possibly by using a laser- or plasma cleaning process.

To get coloured bitmaps following steps are necessary:

- check image and the colours it contains
- find as much as possible pen parameters that result in different colours when they are used for marking on your target material; they should include as much of the colours that are contained in image to be marked
- calibrate your monitor so that the colours displayed there are as close as possible to the real colours
- walk through all pens where you found parameters for coloured marking results, enable option “Use for coloured scanner-bitmaps” and click the colour chooser “Colour after marking”, there select a colour that is as close as possible to the marking result of this pen (that's the step where the calibrated monitor is required)
- close pen dialogue
- import your coloured image using menu Project → Import
- select option “Use as coloured scanner bitmap” in image import dialogue; when this option is not enabled, you don't have enough pens defined for usage in coloured bitmaps
- on top left corner of import dialogue a preview of the image can be seen that makes use of the colours defined by the pens; now the image can be manipulated using the sliders for Brightness, Gamma, Contrast, Correction Red, Correction Green and Correction Blue to improve the result

Now after clicking OK the image is imported. It is processed in a way that maps the existing coloured pens to closest colours in image. During marking, such an image is processed in several cycles, one for every colour/pen that is contained.

## 5.8 Construction Of Geometry

Comparing to other CAD applications that offer basic primitives with huge amount of options and parameters the concept of BeamConstruct is different: It offers several basic primitives like lines, circles and others that contain only as less parameters as necessary. To get a wide range of effects and complex geometries, these basic data can be combined with other elements that add new data to them or that modify and manipulate them. Thus several simple and easy to understand elements can be put together to complex geometries.

BeamConstruct offers following element types that can be used for these combinations:

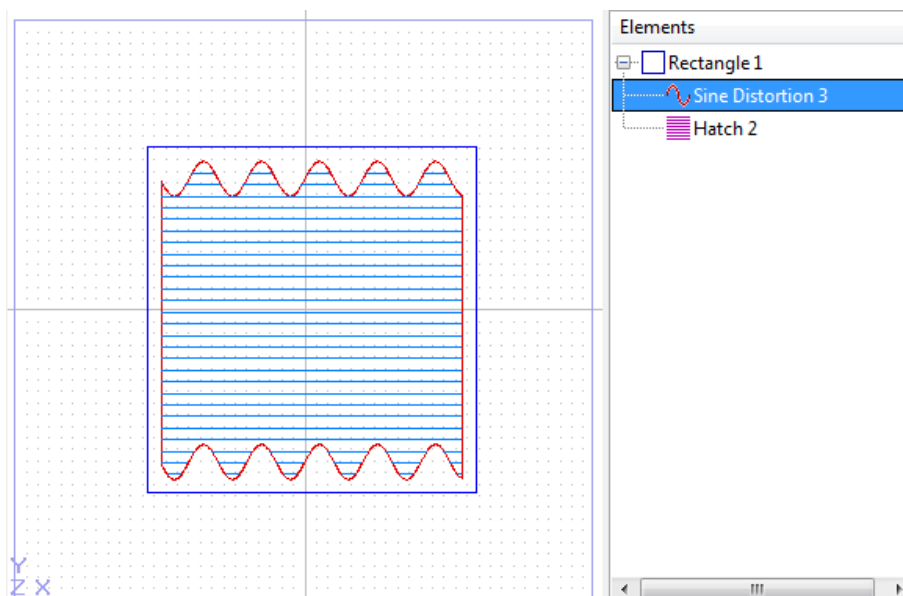
- Primary Geometry – such an element is always the starting point and offers a basic, primitive geometric element (like a rectangle, triangle, text or others like this); this element can be combined with several other elements like:
- Additional Geometry – this elements uses a previously defined geometry and adds some more geometry to it
- Post-processing Tool – this element modifies existing geometry
- Input Element – here an indirect influence to existing geometries is possible, this kind of element gets data from an external source and puts it into Primary Geometry elements that can use these data for their own geometry (like the text or barcode element)

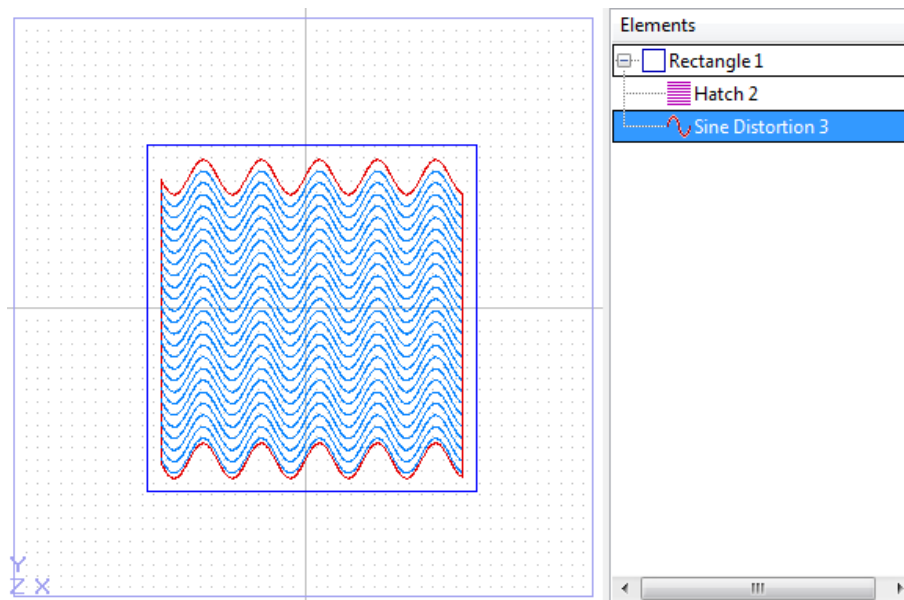
Following rules apply to these elements:

- a Primary Geometry element can have several child elements but none of type “Primary Geometry”
- Additional Geometry and Post-Processing Tools can exist several times as child of a Primary Geometry element, but they influence only the state of the geometry like it exists before their own position
- an Input Element can exist exactly once for a Primary Geometry element and only for such elements that really are able to handle input data)

The second rule can be made explained better with an example: as Primary Geometry a rectangle is drawn. It is modified with a “Sine Distortion” Post-processing Tool which changes the borders of the rectangle to have a sine waveform. Next an additional element “Hatch” is used which adds lines to the geometry that fill the rectangle

In this case the sine distortion is applied to the rectangle only (because it is located before the sine-element) but not to the hatching lines (because they are added after the sine distortion takes place. So the hatching lines stay straight:



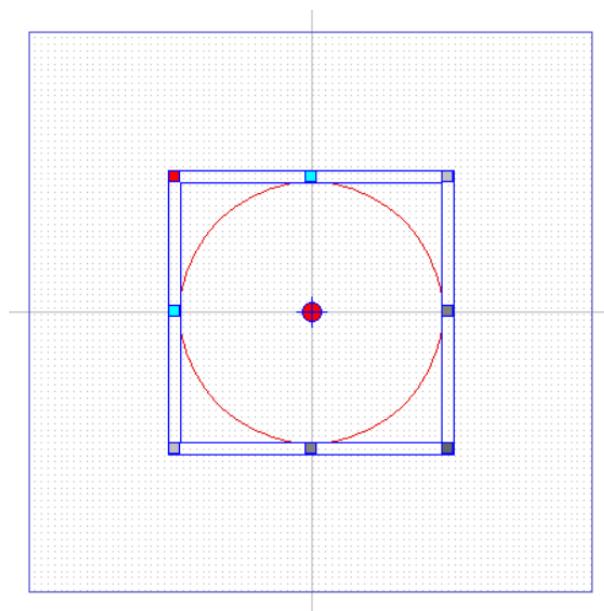


The situation becomes different when the two sub-elements change their position. When the first child element below of the primary element is the hatcher the resulting hatch lines will be distorted by the following Post-processing Element too. They are no longer straight lines but have the same sine shape like the rectangles border because the sine distortion now applies to the hatcher too: that's caused by the fact that the sine distortion Post-Processing Tool stays after the hatcher:

PLEASE NOTE: for the example explained above the line styles for rectangle and hatch have to be switched to "Connected Lines".

### 5.8.1 The Drawing Area

Within BeamConstruct's main window the drawing area for geometry is located in the middle. There new primary elements can be created or existing geometry can be modified. Whenever an element is selected, it is highlighted by a double-lined, blue border. This border can be used to manipulate the element. When pressing the left mouse button within this border and outside the coloured squares it contains, it can be moved and repositioned. Left-clicking it within one of the squares gives access to other manipulation functions:





- the dark-grey rectangle at the lower right corner provides a resizing-function where the aspect ratio between both visible directions is kept intact
- the medium grey rectangles at the lower or right border of the rectangle provide resizing functionality in one direction only
- the light-grey rectangles in upper right and lower left corner allow to rescale the element in every direction and without keeping the current aspect ration between the two visible directions
- the cyan rectangles at the left and upper border can be used to slant the geometry in one of the visible directions
- the red rectangle in upper left corner can be used to rotate it
- the red circle with the blue cross-hairs specifies the centre position which is used for rotation operations; it can be dragged to any other position resulting in a different rotation centre. To set this centre point to a specific position manually by entering coordinate values, please check out the related function in “Geometry” tab-pane (for details please refer to section “5.8.2.2 The Geometry Tab-Pane” below)

How and by which value the elements are changed can be seen within the panels on the main windows left hand side as described in following section.

Beside the possibility to drag and move the element by holding it in its outer, double-lined border it also can be moved via the cursor keys when Control is held down:

- holding Control- and Shift-key and hitting cursor up/down/left/right moves the selected element by 5 mm into the given direction
- holding the Control-key and hitting cursor up/down/left/right moves the selected element by 1 mm into the given direction
- holding Control- and Alt-key and hitting cursor up/down/left/right moves the selected element by 0,1 mm into the given direction

PLEASE NOTE: The manipulation possibilities described above apply to the geometry that is currently selected. When a primary element is selected the changes apply to its geometry, possibly existing additional or Post-processing Tools are re-calculated according to these changes. Whenever additional geometry within a Primary Element is selected and manipulated, this has no feedback to its parent Primary Element, in this case the Additional Geometry might be changed without any relation to the parents geometry and therefore in a way that is not desired.

The part of the current project that is visible within the drawing area can be changed too: when no element is selected the whole area can be moved by holding down the left mouse button and dragging it into the direction where the complete view has to be moved to. Beside of that there are zoom-buttons available within the tool-bar:

- Zoom In – zooms into the drawing area so that more details become visible
- Zoom Out – show a bigger area but less geometry details
- Zoom To Working Area – changes zoom and position of the current view so that the full working area is visible in the maximum size that is available in the drawing area
- Zoom To Selected – changes zoom and position of the current view so that the currently selected element is visible in the maximum size that is available in the drawing area

When one single element is selected that contains some geometry, a special point editing mode can be enabled to modify the single points this element consists of. This mode can be enabled by turning on the tool-bar button right beside the zoom-buttons. Once this button is enabled the appearance of the current element changes, now all editable points are surrounded by a rectangle. These points now can be dragged to other positions. This can be done for every point separately. To drag a group of points together please refer the description of the Geometry Tab Panes table “Coordinates” below.

### **5.8.1.1 Select Elements**

To select existing elements for further editing there exist different possibilities:

- left-click the geometry of a single element in drawing view → the element is selected, when it is part of a group, the whole group is selected
- left-click the name of a single element in element tree → the element is selected
- left-click the geometry of several elements in drawing view while Shift- or Ctrl-key is held down → the elements are selected, when one of them it is part of a group, the whole group is selected
- left-click the name of several elements in element tree while the Ctrl-key is held down → the clicked elements are selected, clicking an element again un-selects it
- left-click the name of two elements in element tree while the Shift-key is held down → all elements which are in range from first to the second clicked element are selected
- draw a frame around several elements in drawing view while the Alt-key is held down → all elements which are touched by the drawn frame are selected, when one of them is part of a group, the single element is selected but not the whole group

### **5.8.1.2 2D and 3D Editing**

Within most scanner controller parameters (please refer to the description of the project settings above) it is possible to choose between a 2D operation mode (two-axis mode) and a 3D operation mode (three axes controlled by the scanner controller). This option depends on the capabilities of the used scanner controller and influences the behaviour of BeamConstruct's drawing area. When a two-axis mode is selected, only the two axes X and Y can be modified within BeamConstruct and the drawing area offers manipulation possibilities for X (horizontal direction) and Y (vertical direction) only.

As soon as 3D mode is enabled, the drawing area is extended by some semi-transparent buttons that are shown directly within the drawing areas upper left corner. These buttons can be used to switch the current view and the axes that are visible within this view:

- top view – the default view for a laser marking system from top where the laser source is located above the working piece, here X/Y are shown in horizontal/vertical direction of the drawing area
- front view – view from front where X/Z are shown in horizontal/vertical direction of the drawing area
- right view – view from right side where Y/Z are shown in horizontal/vertical direction of the drawing area
- 3D view – here all three dimensions of the current project are visible at the same time within a three-dimensional, foreshortened view; comparing to the other views here no elements can be created and edited but it is possible to rotate the view by moving the mouse while holding down the right mouse button
- split view – when this view is enabled, all four views described above are shown at the same time within a split view. This mode requires more hardware resources but makes editing easier because all three views (top, front, right) can be edited simultaneously.

### **5.8.1.3 Drawing Area with 3D surface**

Normally the drawing area – which is related to the marking area – is assumed to be flat. For special applications and when a 3D scanhead is used, here also a three-dimensional surface can be defined where the 2D vector data can be projected onto. This allows to mark on irregular surfaces while keeping the focus of the laser.

To make use of a marking area with a 3D surface, following steps are necessary:

- import a 3D model (via menu item “Import”) which contains the 3D surface
- in import dialogue, select the option “Use as ground shape”
- now a 3D Mesh object is part of the whole project, which is a bit special: it can not be marked and it can not be selected by left-clicking it in the Drawing Area, this can be done only via element tree
- select the ground mesh in element tree and place it within the Drawing Area as required

- recommended: assign this 3D mesh to a specific layer which then can be hidden via the “Layer” tab pane – this makes it easier to edit marking geometries as the ground object is not always shown
- create/edit marking data as usual (for details about the different primary geometric elements and their purpose, please refer to section 5.8.4 Primary Geometry Elements below)
- as soon as the geometry is complete, select all elements to be marked on top of the ground surface, right-click the selection and choose function “Project onto ground”
- now a dialogue opens which asks for the minimum vector length the geometries have to be split into prior to projecting them to the ground object: the smaller this value is, the more accuracy the projection results become but the more marking data are generated and the more computer resources are consumed

After that the marking vector data follow the shape of the ground object in Z.

## 5.8.2 The Property Tab-Panes

The main window of BeamConstruct contains some special tab-panes on its left hand side. These panes give access to different parameters and values, they can be used to influence geometries in a very exact way.

### 5.8.2.1 The tab-pane Of The Currently Selected Element

The very left tab-pane is a dynamic one, it changes its contents depending on the element that is currently selected. As long as no element is selected this panel is empty, no parameters can be changed here. These element-specific values that become visible as soon as something is selected within the drawing area or within the element tree on the windows right hand side are described in detail below together with the different elements.

In case more than one element or in case a simple group element is selected this tab-pane is empty too.

### 5.8.2.2 The Geometry Tab-Pane

This panel can be used only when an element is selected that contains some geometry. Thus it does not offer any functionality as long as no element or as long as a geometry-less Post-Processing Tools is selected.

Within this panel geometry-related values are displayed and can be changed. When 3D mode is enabled, the number of available and usable options and parameters depends on the currently selected view, in 2D mode only these values and options can be changed and used that belong to a flat, two-dimensional project that only extend in X and Y direction:

- Mirror X, Mirror Y and Mirror Z buttons – when pressing one of these buttons the currently selected geometry is mirrored in X, Y or Z direction; pressing the same button twice results in no changes (element is mirrored and then mirrored back to original state in this case)
- Centre X, Centre Y and Centre Z buttons – these buttons reposition the selected geometry within the current working area, they centre it in X, Y or Z direction
- Scale – these input fields show the current scaling factor comparing to the original state of the geometry for X, Y and Z direction, a value of 1.0 means the geometry is not scaled, values smaller than 1.0 mean it is shrunk in this direction, values greater than 1.0 mean it is bigger comparing to the original size
- Top Left – the upper left coordinates of the selected element, whenever they are changed the element is repositioned without changing its size
- Centre pt – the centre coordinates of the selected element, whenever they are changed the element is repositioned without changing its size. Different to the other rows, the text in front of the centre

coordinate input fields is a button. When this button is pressed, a dialogue is opened which shows the coordinates of the centre point of the currently selected element. When these centre coordinates are changed within this dialogue, only the position of the centre point is changed but not the position of the element itself. This is an operation comparable to the possibility to drag the centre point of an element within the drawing view (please check out section “5.8.1 The Drawing Area” for more details).

- Bottom Right – the lower right coordinates of the selected element, whenever they are changed the element is repositioned without changing its size
- Size – the total width, height and depth of the element, when it is changed the geometry is scaled in X-, Y- or Z-direction, the related values for scaling
- Rotation – the angle in unit degrees the element is rotated with, here the rotation can be changed which may result in changes of other parameters depending on the shape of the used geometry
- Coordinates – this list contains the coordinates of all points of the currently selected geometry, here they can be changed to re-position a single point of it; selecting a point within this list also affects the highlighted element within the drawing area: there the selected point is surrounded by a separate, blue box which can be used to drag this point, thus the geometry can be edited using the mouse. Within this table of coordinates a single point or groups of points can be selected. This can be done by left-clicking into the row of the point that has to be selected. Additional points can be chosen by selecting additional rows of points while the shift or control key is held down. In case more elements are selected they can be modified within the drawing area by dragging them to other positions, here all highlighted points are moved together.  
Comparing to this a point editing mode is available that gives the possibility to highlight and edit all points of an element separately. For more details about this mode please refer above.

For the parameters described here the same is true as described for the manipulation functionalities within the drawing panel: When the changes are applied to a primary element, they influence the subsequent elements too. When an additional element is selected there is no feedback to its parent primary element, the changes may override the parents geometry.

### 5.8.2.3 The Element Tab-Pane

Comparing to the Geometry-panel this one contains more global parameters of the selected element. Depending on the type of selected element only some of the parameters are accessible:

- Type – this is an informational field and therefore read-only, it displays the type of the element
- Name – here the name of the element is shown and can be changed; this name is auto-generated when the element is created and shown within this panel and within the element-tree on the main windows right hand side; for some operations out of e.g. BeamServer or own applications it may be necessary to have unique names here.
- Pen – a pen is a definition about how the related geometry has to be processed on the target system; here up to five different pens as well as the number of repeated marks this pen has to be used with can be assigned to the current geometry so that it is processed using these specific pen parameters (please refer above for a description of the pen parameters)  
An element always needs to have at least one pen, so the first one is mandatory and can't be neither disabled nor turned off.  
All following pens below of this first one are optional: they can be enabled by checking the related box. When an element makes use of more than one pen, it is processed repeatedly using each of the provided pens. Thus it is possible to mark the same element with several pens without the need to duplicate its geometry.  
The number input field right behind every of these pen-definition-rows specifies an additional type of loop: this value specifies, how often a geometry has to be marked with the related pen. A value of 0 means, the geometry is marked exactly once with this pen.
- Head – in case of a multihead environment (as described in “5.16 Multihead Operations”) there is the possibility different controller cards are able to process the same geometries. Here the software typically decides for its own and based on the current global settings (as done in tab-pane “Multihead” as described at “Error: Reference source not foundError: Reference source not found”)

which controller is responsible for which part of the geometry. For these cases, where a geometry has to be assigned to a controller card (aka “head”) directly, this selection box can be used. Here a head can be selected to process the geometries of the related element. When no head is selected but option “Automatic” is chosen, the default, automatic behaviour as defined in general project settings is used for this element. Here “Automatic” is the default behaviour and should be used unless an other selection is necessary.

- Alignment – in general pen-settings there is one option to align the laser/the milling tool at the inner or outer side of a polygon (please refer to “5.7.4 Laser Pen Parameters” for details). This gives the possibility to cut elements in exactly the size specified by these geometry by taking the cutting width of the laser/tool into account. Instead of defining such an alignment globally via the pen settings, using this selection list it is possible to align a single element in the same way. Means when a different value than “From pen settings” is chosen here, the “Align spot to vectors” parameter out of the used pen is ignored but the local definitions as per these settings are used.
- Layer – BeamConstruct supports several drawing and displaying layers that can be used to separate parts of the whole project from each other and to make it easier to manage complex projects. These layers can be hidden and shown depending on which has to be visible. Using the combo-box provided here the selected element can be assigned to one of the available layers. For more information please refer the section about the layer tab-pane below.
- Process element – this check-box decides if the related element has to be processed or not, when it is not checked the related element acts as some kind of placeholder only, it is not used by the target system for processing the working piece
- Mark contour first – on a standard marking process first the additional element (the “hatch”) is marked and at the end the contour is appended. When this option is set for a element which itself is the contour (e.g. a primary element), this order is inverted, then first the contour is marked followed by the hatch
- Set invisible – turns off the geometry in the Drawing View; when this option is set for an element, it is no longer visible but the related geometries are still marked. In this state it can be selected and modified via the Element Tree only
- Trace data – when this option is set, the name, and when available, unique data of this element are used for traceability purposes. This means, when a marking cycle was performed successfully, the related Smart Interface message or trace log entry contains information about this element too. These information consist of the name of the element. When this option is enabled for an input element, beside the name also the possibly unique input data are added to the trace data. For more information please refer to section “5.22 Smart Interface” (when the trace data have to be provided via the Smart Interface) or to section “5.5 Project Configuration” option “Trace log” (in case the trace data have to be written into a log file)
- Count X/Y/Z and Distance X/Y/Z via “Reference copies” – these parameter fields belong to each other and can be used to create arrays of geometry without the need to duplicate them; here it can be defined how often they have to be repeated in X, Y and Z direction and which distance has to be set between them;  
Below of these array-parameters an option “Fade out references” can be found. When this option is set, only the original geometry of the array is drawn, the referenced copies are replaced by placeholders showing just the outline. This option can be useful in cases where the referenced geometries consist of huge, complex vectors and/or when a weak graphics card is used, so that the whole application is slowed down just by these reference copies. When set, the load on the graphics card is lowered, resulting in a much more responsive reaction on the user interface.
- Number of vectors – this is an informational element and can't be changed directly, it displays the number of vector lines that are used to create the selected element
- Repeat marks – using this parameter it can be specified how often marking of a single element has to be repeated during a single marking cycle; here a value of 0 means, it has not to be repeated, thus it will be marked exactly once
- Skip marks – when this is set to a value greater than 0, the related element is no longer marked on every process cycle but skipped for the given number of times; after this number has elapsed, it is marked once and skipped afterwards again

- Skip offset – this parameter belongs to "Skip marks" and specifies an offset at what mark/skip counting has to start; so with both parameters elements can be created that are marked interleaved:
  - Element 1 with skip marks = 1 and skip offset = 0
  - Element 2 with skip marks = 1 and skip offset = 1
  - results in two elements that are marked alternating, each exclusively
  
  - Element 1 with skip marks = 2 and skip offset = 0
  - Element 2 with skip marks = 2 and skip offset = 1
  - Element 3 with skip marks = 2 and skip offset = 2
  - results in three elements that are marked alternating, each exclusively
- Total line length – this field contains an information too, here the total length of the geometry is shown; this value takes the "on"-lines into account only, "off"-lines (movements with the tool turned off to go to the beginning of the next "on"-line) are ignored here
- Process time – here the time is displayed that is needed to process the currently selected element; this value is calculated out of the length of "on"-lines, the length of "off"-lines between them and the speed and delay information of the pen that is assigned to this element

#### **5.8.2.4 The Layer Tab-Pane**

Every element can be assigned to exactly one layer. These layers are in no way related to the processing of the data within the target system, but they are a useful utility that makes it easier to handle complex projects.

Within this tab panel two things can be done: layers can be renamed according to their meaning and they can be turned on and off.

When the box in front of a layer is checked, all elements that are assigned to this layer are drawn within the drawing area. Layers that are not checked are not visible within the drawing area, they are hidden and therefore can't be selected or edited.

When a marking process is started out of BeamConstruct the layer visibility states do not have an influence on the marking result. Means also these layers that are currently hidden will be marked.

#### **5.8.2.5 The Motion Tab-Pane**

In tab panel Motion it is possible to drive configured motion axes manually and independent from any motion elements in current marking project. Please note: when there are some marking elements used directly within the project, it has to be taken into account that the user may have changed the position of the axes via these manual functions. Thus at least the first motion element of a project should perform a movement to a defined, absolute position or should reference its position.

Within the motion tab-pane following functions are available:

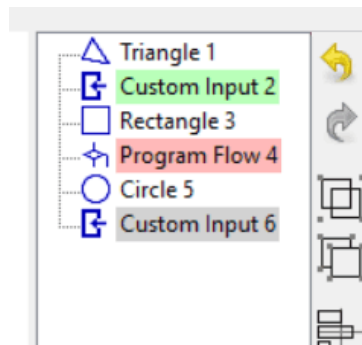
- OUT2 / OUT3 – specifies which of the two possible motion controllers has to be used
- Speed – specifies the speed a manual motion operation can be performed with;  
PLEASE NOTE: when the speed value given here is bigger than the maximum speed of the related axis, this value is clipped internally, means the axis moves with it's maximum speed which may be slower than the nominal speed entered here
- movement value – the value that can be entered in middle between the arrows specifies the relative movement distance or the absolute motion position to move to (depending on the arrow button that was pressed)
- Double arrows – these buttons cause a movement as long as the button is pressed
- Single arrows – these buttons cause a single movement via the distance or angle specified with the number input field in the middle, here the time the button is pressed does not influence the motion
- Arrow to dot – when this button is pressed an absolute movement is performed, the number input field in the middle between the buttons specifies the target position to move to, the time the button is pressed does not influence the motion

- Ref – start referencing of the related axis

For planar movement axes the images of the buttons can be changed within the motion axis configuration panel of the general project settings. Depending on the real movement direction (left/right, up/down, in/out) a different symbol can be shown here in order to make it easier for the operator to decide which direction a motion will use. The same is true for the axis names, they can be configured via the axis alias value in general project settings.

### 5.8.3 The Element-Tree

On the main windows right hand side a tree is shown that represents the current projects structure. It consists of a list of primary elements and their (optional) child elements. During processing on the target system the elements are typically processed in the same order as they are listed there from top to bottom (including their children). The background colour of each item signalsises the way processing is done after an element:



- white background (aka default background of the tree) – processing always continues with the next element in the tree-structure; this is the default behaviour of most of the elements and the default behaviour typically needed for standard marking projects
- green background – processing may be restarted, means the project is executed again from its beginning; if this restart-condition really applies, depends on the elements behaviour and its configuration
- red background – processing may be stopped, means project is cancelled at this point; if this stop-condition really applies, depends on the elements behaviour and its configuration
- grey background – processing may continue at a different element of the project, means the application jumps within the current tree; if this go-to-condition really applies, depends on the elements behaviour and its configuration

Every primary element may contain several Additional Geometry elements and several Post-Processing Tools. They are listed as child nodes of their Primary Geometry parents within this tree.

All these primary and child nodes can be edited within this tree via a context menu. This menu can be opened by selecting a node within the tree and clicking the right mouse button. Then following functions can be chosen:

- Group elements – while holding down the Shift- or Ctrl-key it is possible to select a range of elements within the entity tree; in case more than one Primary Geometry element is selected it is possible to put them into a group using this function
- Ungroup elements – this item can be used when a group element is selected, it resolves the group, pulls all element out of it into the hierarchical level where the group existed before
- Resort elements – this option is active only when a group with some sub-elements is selected. It gives the possibility to automatically re-order the elements within this group. For a detailed description of this function please refer below

- Project onto ground – this function is part of the possibility to have a 3D working area surface (as described in section “5.8.1.3 Drawing Area with 3D surface ”). It changes all selected geometries in Z-direction in a way, so that they are located on the ground surface
- Delete Element(s) – deletes an element completely; in case this is applied to a primary element all its sub-elements are removed too
- Duplicate – the currently selected Primary Geometry element is copied and inserted at a position that is slightly shifted according to the original element
- Move to top – the selected element is moved to the top position within its same hierarchically level (means a child of a primary element will stay a child of it but it will become the first one)
- Move up – the selected element is moved one position up but also within the same hierarchically level
- Move down – the selected element is moved one position down within the same hierarchically level
- Move to bottom – the selected element is moved to the bottom position within its same hierarchically level (means a child of a primary element will stay a child of it but it will become the last one)
- Import 3D model – this option is active only when a single Sliced 3D Mesh Group element is selected, it gives the possibility to import an additional 3D Mesh into this Sliced 3D Mesh Group; after successful import this Sliced 3D Mesh Group keeps two more 3D Meshes and creates slices that consist of all of their 3D data, the 3D Meshes which can be positioned within the working area independent from each other
- Clean between slices – this option is active only when a single Sliced 3D Mesh Group element is selected; when this option is set, such a 3D Slice Cleaner element is generated on construction of the Sliced 3D Mesh Group. For more information about this type of element which can be used for cleaning purposes between slices, please refer to section “5.10.4 3D Slice Cleaner”
- Auto-arrange – this option is active only when a single Sliced 3D Mesh Group element, when a group or when several primary elements are selected, it gives the possibility to arrange several 3D Meshes or child elements within a group; for a detailed description of the auto-arrange function please refer below
- Merge 3D meshes – when several 3D slice groups are selected, this option can be chosen to merge their 3D meshes into one element. After this operation has been finished, all the selected 3D slice groups are combined into one containing all the 3D meshes as one single object
- Collapse all – closes all nodes in the current element tree; when all nodes are collapsed and the whole tree is selected (with shift-key held down) only the top level primary elements are selected but not their sub-elements
- Expand all – opens all nodes in the current element tree

### **5.8.3.1 Auto-Resorting of elements in groups**

A group can hold several sub-elements as children. The order these elements are marked is specified by the order they are added to their parent group. This order can be changed by manually moving single elements up or down within this group using the related context menu functions within the element tree. Alternatively it is possible to let BeamConstruct resort these elements automatically. Different to the rearrange-function as described below, this one does not change the geometric position of the elements, but their order.

To resort the elements, the group has to be selected and right-clicked in element tree, then menu item "Resort Elements" has to be selected. Now a dialogue opens which provides options to change the order of the elements within their parent group. This is done via the configuration possibilities within the dialogue:

- Skip fixed size – the current order of elements is changed in a way where every nth element is used for the new order. This is done repeatedly until all elements are used. Here value “n” can be specified by the related input field. For an example: a value of 2 will first take every odd-numbered element and then all remaining (even-numbered) elements for the new order.



### 5.8.3.2 Auto-Arrangement of elements in groups

For a description of auto-arrange function of 3D Meshes in Sliced 3D Mesh Groups please refer to the related section below.

A group can hold several sub-elements as children. The location of these elements within the drawing view can be positioned with the standard functions in tab-pane "Geometry" in order to place them at suitable positions in working area. Alternatively it is possible to let BeamConstruct arrange these elements automatically. Different to the resort-function as described above, this one does not change the marking order of the elements, but their position.

To rearrange the elements, the group has to be selected and right-clicked in element tree, then menu item "Auto-Arrange" has to be selected. Now a dialogue opens which provides options to arrange the elements in X- and Y-direction and to align them at a specific Z-height. This is done via the configuration possibilities within the dialogue:

- Arrange in X and Y – when enabled, the sub-elements are arranged on working area using the following parameters
- Distance to borders – specifies the minimum distance the child-elements of the group need to have to the border in both, X and Y direction of the working area
- Distance between elements – specifies the minimum distance the children of the group need to have between each other
- Arrange in Z – when enabled, the elements are aligned at a specific height using the following options and parameters
- Align geometries at top or bottom – here it can be selected if all the elements need to be aligned at their upper or lower position
- Arrange in Z – this is an option which can be enabled when elements have to be top or bottom aligned at a specific, absolute height; when it is set, the height has to be specified in unit mm; in case this option is not set but arrangement in Z is enabled, all sub-elements of the group are aligned at the highest/lowest top/bottom position of that element, that is located at the highest/lowest position (according to its outlines top/bottom border)

### 5.8.4 Primary Geometry Elements

To create a new primary element following steps have to be performed:

1. select the related button within the tool-bar, all primary elements use blue tool-bar symbols
2. click into the working area to position the element; depending on the geometry that is created by the related function it can be drawn now with some additional clicks into this area
3. now the new element is selected and its parameters are shown within the first, element-specific tab-pane on the main windows left hand side

When the parameters of such an element are changed within its tab-pane, all following elements of the same type use exactly these parameters as default values.

Following the available elements, the steps to create them within the working area and their parameters are described.

#### 5.8.4.1 Dot

This is the most simple element that is possible. When it is selected for drawing within the tool-bar, the first click into the drawing area creates a single point at the left-clicked position.

This element does not offer any parameters, so there are no input fields available within the element-specific tab panel as soon as such a dot is selected.

Within the options the dot can be configured:

- Style – specifies the style of the created point, here Dot is a simple one that consists of a single coordinate and has no size or Custom for a dot as configured within the global parameters (please refer to section 5.7.1 above)

#### 5.8.4.2 Line

When drawing a line two left-clicks are necessary within the drawing area: the first click specifies the starting position of the line, the second one the end position. It is not necessary to keep the mouse pressed between these two clicks.

Drawing functionality can be extended by pressing one of the keys "Shift" or "Ctrl" after specifying the starting point of the line (first click): when Shift is held down, a horizontal line is drawn, and vertical deviations of current mouse position are ignored. When Control is held down, a vertical line is drawn, and horizontal deviations of current mouse position are ignored.

After drawing or selecting such a line following parameters are available to manipulate them:

- Line Style – the style the line is created with, here following options exist:  
Continuous: the geometry consists of exactly one starting and one end point where one single line is drawn between  
Connected Lines: the geometry consists of several short lines including additional points between the start and end point, all these lines are connected  
Dashed: the geometry consists of several short lines including additional points between the start and end point, every second line is drawn that results in a dashed line style  
Dotted: no lines are drawn, the geometry consists of a set of not connected dots  
Dotted (Custom): no lines are drawn, the geometry consists of a set of not connected points in custom dot style as configured within the project settings (please refer to section 5.5 Project Configuration above)
- Distance – whenever a non-continuous line style is selected, this distance value specifies the length of the additional points that form the selected line style
- Expand By Dot – expands the line according to the chosen distance-value by one dot (or one line segment)
- Remove Last Dot – removes the last dot or line segment

#### 5.8.4.3 Triangle

To draw a triangle all three corners have to be specified separately by single left mouse clicks for each. First the starting point is defined, with the next click an other corner of the triangle is specified and the last one completes the triangle by setting the third corner of it.

Drawing functionality can be extended by pressing one of the keys "Shift" or "Ctrl" after specifying the first corner point of the triangle (first click): when Shift is held down, a horizontal line is drawn next, and vertical deviations of current mouse position are ignored. When Control is held down, a vertical line is drawn, and horizontal deviations of current mouse position are ignored.

Within the triangles property tab-pane following options are available:

- Line Style – the style the triangles lines are created with, here following options exist:  
Continuous: the geometry consists of exactly three points that are connected by three completed lines  
Connected Lines: the geometry consists of several short lines connecting the three corners including additional points, all these lines are connected and therefore form a closed shape  
Dashed: the geometry consists of several short lines including additional points between the three corners, every second line is drawn that results in a dashed line style  
Dotted: no lines are drawn, the geometry consists of a set of not connected dots  
Dotted (Custom): no lines are drawn, the geometry consists of a set of not connected points in custom dot style as configured within the project settings (please refer to section 5.5 Project Configuration above)

- Distance – whenever a non-continuous line style is selected this distance value specifies the length of the additional points that form the selected line style

#### 5.8.4.4 **Rectangle**

When drawing a rectangle the upper left and the lower right corner of it are specified by two separate mouse clicks. The resulting rectangle will always be parallel to the coordinate axes.

Drawing functionality can be extended by pressing the "Shift"-key after specifying the first corner point of the rectangle (first click): when this key is held down, a square is created instead of a rectangle. Horizontal distance from first point specifies the side length of the square.

Within the rectangles property tab-pane following options are available:

- Line Style – the style the rectangles lines are created with, here following options exist:  
Continuous: the geometry consists of exactly four points that are connected by four completed lines which form a rectangle  
Connected Lines: the geometry consists of several short lines connecting the four corners including additional points, all these lines are connected and therefore form a closed, rectangular shape  
Dashed: the geometry consists of several short lines including additional points between the four corners, every second line is drawn so that it results in a dashed line style  
Dotted: no lines are drawn, the geometry consists of a set of not connected dots  
Dotted (Custom): no lines are drawn, the geometry consists of a set of not connected points in custom dot style as configured within the project settings (please refer to section 5.5 Project Configuration above)
- Distance – whenever a non-continuous line style is selected this distance value specifies the length of the additional points that form the selected line style
- Nominal width – this is the width of the rectangle (in X-direction) as done by this primary element; so this value does NOT reflect possible scale factors done in "Geometry"-tab-pane, the real geometry width is shown only there
- Nominal height – this is the height of the rectangle (in Y-direction) as done by this primary element; this value does NOT reflect possible scale factors done in "Geometry"-tab-pane, the real geometry height (after all transformation operations have been applied) is shown only there
- Nominal depth – this is the depth of the rectangle (in Z-direction) as done by this primary element; this value does NOT reflect possible scale factors done in "Geometry"-tab-pane, the real geometry depth (after all transformation operations have been applied) is shown only there
- Corner radius – when this value is greater than 0 and when the rectangle is large enough, it is created with rounded corners
- Segments – when the rectangle is created with rounded corners, this value specifies of how much segments all these corners consist and therefore how smooth they are

#### 5.8.4.5 **Circle**

This primary element can be used to create circles and arcs. Starting point is always a circle that is drawn in two steps: the first left mouse click into the drawing area specifies the centre point of it, the second one specifies the radius (via its distance from the first point).

Next the circle can be modified further:

- Line Style – the style the circle is created with, here following options exist:  
Continuous: the geometry consists of several short, fully connected lines that form a closed circle; for this mode the parameter "Segments" specifies how much of these short lines are used to create this circle, the more lines are used the more smooth the circle becomes  
Connected Lines: the geometry consists of several short lines that form the circle, comparing to the previous option here the parameter "Distance" influences the length of these lines and therefore the resolution; here the shorter the lines are the more segments this circle consists of and the more

smooth it becomes

Dashed (Arcs): the circle consists of dashes where the dashes are not straight lines but follow the shape of the circle and therefore are round

Dashed (Straights): the circle consists of dashes where the dashes do not follow the shape of the circle but are straight lines

Dotted: no lines are drawn, the geometry consists of a set of not connected dots

Dotted (Custom): no lines are drawn, the geometry consists of a set of not connected points in custom dot style as configured within the project settings (please refer to section 5.5 Project Configuration above)

- Distance – this parameter decides how long the resulting lines of the chosen line style have to be, it is not valid for the continuous line style where the length would be always equal to the circumference
- Radius – the radius of the circle
- Second radius – this is an optional parameter. When enabled, a radius value can be entered here which is used to create a second circle. Both, the primary and the second radius can be used to form specific shapes like rings.
- Segments – here it can be chosen how much segments the circle has to consist of, this parameter influences the accuracy of the resulting circle; for the line style “Dashed (Arcs)” it influences the smoothness of the resulting arcs while the “Distance”-parameter influences their length
- Start Angle and End Angle – these two parameters can be used to change the circles starting and end angle, it can be used to draw arcs (in case the resulting total angle is smaller than 360 degrees) or to let a circle be drawn several times (in case the resulting total angle is a multiple of 360 degrees)
- Grow: this parameter is valid in 3D mode only; it defines how much the circle shall grow in the third direction per every rotation of it. Here “third direction” does not necessarily mean Z-direction, this depends on the view the circle is drawn in. So in case the right side view is enabled the circle extends in Y and Z direction, resulting from that the third direction where the grow parameter applies to is the X direction.
- Layered Sphere: in 3D mode this option can be used to create a sphere that consists of circles of different diameters that form a sphere; to use this option the Grow value has to be set which specifies the distance between the circles that form the sphere
- Draw CCW – normally the circle has a clockwise orientation, means it is drawn to right direction; when this check box is set, the drawing direction is reversed

#### 5.8.4.6 Star

This element is a special variant of a circle, it consists of two radii where every second point the element consists of, belongs to an other radius. Resulting from that stars can be drawn and modified by its parameters. Such a star is drawn in two steps: the first left mouse click into the drawing area specifies the centre point of it, the second one specifies the first radius. While drawing the star the second radius is set automatically to value smaller than the first radius.

After the star is drawn it still can be modified interactively. Here every second point (the ones that belong to the second radius) is marked by a small, blue cross. This cross can be dragged with the mouse in order to change the second radius independent from the first one and in order to apply an offset to the angle of these points.

Next the star can be modified via its options:

- Line Style – the style the circle is created with, here following options exist:
  - Continuous: the geometry consists of single, continuous lines
  - Connected Lines: the geometry consists of several short lines including additional points, all these lines are connected
  - Dashed: the geometry consists of several short lines including additional points, every second line is drawn that results in a dashed line style
  - Dotted: no lines are drawn, the geometry consists of a set of not connected dots
  - Dotted (Custom): no lines are drawn, the geometry consists of a set of not connected points in

custom dot style as configured within the project settings (please refer to section 5.5 Project Configuration above)

- Distance – this parameter decides how long the resulting lines of the chosen line style have to be, it is not valid for the continuous line style
- Radius 1 – the first radius that is valid for all odd-numbered points
- Radius 2 – the second radius that is valid for all even-numbered points
- Spikes – here it can be chosen how much spikes (or rays) the star shall have,
- Angle offset – here an offset can be specified for the points that are assigned to the second radius in order to create non-regular spikes
- Draw CCW – normally the star has a clockwise orientation, means it is drawn to right direction; when this check box is set, the drawing direction is reversed

#### 5.8.4.7 *Spiral*

This primary element can be used to create spirals. Starting point is always a spiral that is drawn in two steps: the first left mouse click into the drawing area specifies the centre point of it, the second one specifies the outer radius.

Next the spiral can be modified via its options:

- Line Style – the style the spiral is created with, here following options exist:
  - Continuous: the geometry consists of several short, fully connected lines that form a continuous spiral; for this mode the parameter “Segments” specifies how much of these short lines are used to create this spiral, the more lines are used the more smooth the resulting spiral becomes
  - Connected Lines: the geometry consists of several short lines that form the spiral, comparing to the previous option here the parameter “Distance” influences the length of these lines and therefore the resolution; here the shorter the lines are the more segments this spiral consists of and the more smooth it becomes
  - Dashed (Arcs): the spiral consists of dashes where the dashes are not straight lines but follow the shape of the spiral and therefore are round
  - Dashed (Straights): the spiral consists of dashes where the dashes do not follow the shape of the spiral but are straight lines
  - Dotted: no lines are drawn, the geometry consists of a set of not connected dots
  - Dotted (Custom): no lines are drawn, the geometry consists of a set of not connected points in custom dot style as configured within the project settings (please refer to section 5.5 Project Configuration above)
- Distance – this parameter decides how long the resulting lines of the chosen line style have to be, it is not valid for the continuous line style where the length would be always equal to the circumference
- Inner Radius – the inner, minimum radius of the spiral
- Outer Radius – the outer, maximum radius of the spiral
- Slope – this value defines the steepness of the spiral, so this value influences how strong it rises per rotation
- Segments – here it can be chosen how much segments the outer part of the spiral has to consist of, this parameter influences the accuracy of the result; for the line style “Dashed (Arcs)” it influences the smoothness of the resulting arcs while the “Distance”-parameter influences their length
- Smoothing Factor – normally a spiral is generated out of short lines where the length is calculated out of the number of segments and the current radius. This leads to an adaptive line length with as less as much data and an as fast as possible, resulting process time. But in some cases this may lead to spirals that are not very smooth but angular at their inner radius. Here the number of segments could be increased (what would lead to a higher amount of data used because of smoother corners also at curves where it would not be possible) or the smoothing factor can be set to values greater 1.0: this changes the calculation of a spiral in a way where the inner line length becomes shorter and therefore more smooth while not using more data than necessary. So the more smoothing factor is taken for the calculation the smaller the radius becomes.

- Start Angle – this parameters can be used to change the spirals starting angle
- Grow: this parameter is valid in 3D mode only; it defines how much the spiral shall grow in the third direction per every segment. Here “third direction” does not necessarily mean Z-direction, this depends on the view the spiral is drawn in. So in case the front side view is enabled the spiral extends in X and Z direction, resulting from that the third direction where the grow parameter applies to is the Y direction.
- Draw CCW – normally a spiral has a clockwise orientation, means it is drawn from its starting point to right direction; when this check box is set, the drawing direction is reversed
- Inside To Outside – this option influences the growing direction, when the check box is set, drawing of the spiral starts at the inner side and ends outside; in case it is not set it starts at the outer side and ends at the inner

#### 5.8.4.8 Polygon

A polygon consists of an undefined number of edges (but requires at least three). Drawing it can be done by left clicking at the position where a corner between the connecting lines has to be positioned. This can be done several times, the process of drawing this polygon can be finished by right-clicking the mouse. Then the last corner and the first position where the user left-clicked are connected by a line.

Drawing functionality can be extended by pressing one of the keys "Shift" or "Ctrl" after specifying the starting point of the polygon (first click): when Shift is held down for one of the following points, a horizontal line is drawn next, and vertical deviations of current mouse position are ignored. When Control is held down, a vertical line is drawn, and horizontal deviations of current mouse position are ignored.

Following options are available to modify the parameters of the polygon:

- Line Style – the style the polygons lines is created with, here following options exist:  
Continuous: the geometry consists of lines that use exactly one starting and one end point where one single line is drawn between  
Connected Lines: the geometry consists of several short lines including additional points, all these lines are connected  
Dashed: the geometry consists of several short lines including additional points, every second line is drawn that results in a dashed line style  
Dotted: no lines are drawn, the geometry consists of a set of not connected dots  
Dotted (Custom): no lines are drawn, the geometry consists of a set of not connected points in custom dot style as configured within the project settings (please refer to section 5.5 Project Configuration above)
- Distance – whenever a non-continuous line style is selected, this distance value specifies the length of the additional points that form the selected line style
- Polygon Style – here it can be chosen if the last polygon point and the first one are connected by a line (style “Closed”) or not (style “Open”)

#### 5.8.4.9 Bezier Curve

Creation and parameters of this element are very similar to the polygon described above but it does not generate direct connections between the specified base points. Using this element a smooth, round Bezier curve is generated. This curve uses the given base points as creation hint but does not necessarily cross them.

Such a Bezier curve consists of an undefined number of base points but requires at least three of them. Specifying the position of these base points can be done by left clicking within the drawing area. This can be done several times, the process of drawing the related curve has to be finished by right-clicking the mouse.

After that the positions of the base points still can be modified interactively. As long as this element is selected, the real position of the base points are marked by blue crosses. These crosses can be dragged with the mouse. Changing the position of the base points also changes the appearance of the resulting Bezier curve.

Following parameters are available to modify the parameters of the Bezier curve:

- Line Style – the style the polygons lines is created with, here following options exist:  
Continuous: the geometry consists of continuous lines that consist of a number of sub-segments which are specified by the parameter “Resolution”  
Connected Lines: the geometry consists of several short lines of a defined length including additional points, all these lines are connected  
Dashed: the geometry consists of several short lines of a defined length including additional points, every second line is drawn that results in a dashed line style  
Dotted: no lines are drawn, the geometry consists of a set of not connected dots  
Dotted (Custom): no lines are drawn, the geometry consists of a set of not connected points in custom dot style as configured within the project settings (please refer to section 5.5 Project Configuration above)
- Distance – whenever a non-continuous line style is selected, this distance value specifies the length of the additional points that form the selected line style
- Bezier Style – here it can be chosen if the last polygon point and the first one are connected by a line (style “Closed”) or not (style “Open”)
- Resolution: this parameter specifies the smoothness of the Bezier line, the bigger it is the more sub-lines are used to create such a curve, the more smooth it becomes and the more resources are used for it

#### 5.8.4.10 Text

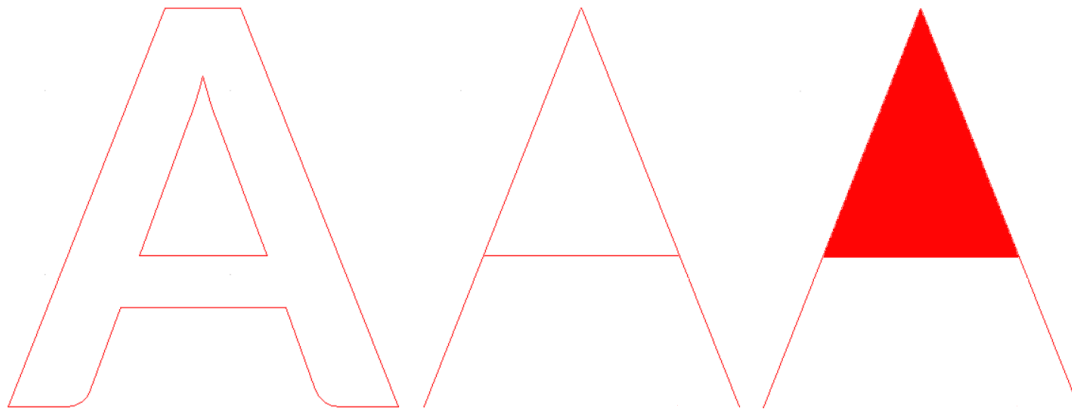
This primary geometry can be used to draw some text using an existing TrueType or special laser font. This element accepts static texts or dynamic values that can be set via an Input Element. When such a project is moved from one computer to another or from one operating system to another it has to be made sure a used font is available on the other system too (this applies to TrueType fonts only since the fast laser fonts are part of BeamConstruct). Otherwise the application tries to find a similar font which might be only somehow similar, but exact positions, sizes and outlines may be got lost in such a case.

Drawing such a text is quite easy: only one single click into the drawing area is necessary to specify the position of the text, there a new geometry is set using a default text. This text can be changed together with some other parameters:

- Line Style – the style the lines the text consists of is created with, here following options exist:  
Continuous: the geometry consists of closed, connected lines with as less as possible points  
Connected Lines: the geometry consists of several short lines including additional points, all these lines are connected  
Dashed: the geometry consists of several short lines including additional points, every second line is drawn that results in a dashed line style  
Dotted: no lines are drawn, the geometry consists of a set of not connected dots  
Dotted (Custom): no lines are drawn, the geometry consists of a set of not connected points in custom dot style as configured within the project settings (please refer to section 5.5 Project Configuration above)
- Distance – whenever a non-continuous line style is selected, this distance value specifies the length of the additional points that form the selected line style
- Text – here a multi-line text can be specified that has to be drawn; a modified text is applied as soon as the text input area is left, either by clicking another element or by pressing the Tab-key
- Unicode – instead of characters here space-separated Unicode values can be entered either as decimal numbers or as hexadecimal numbers with a leading “0x”. On pressing “Return” the entered numbers are converted to text and shown as their character-representation in input area “Text”
- Font Type – here the type of font that has to be used for this text element can be chosen, following options are available to select from:  
TrueType: the standard fonts that are installed on the system can be used. When this option is selected, the combo box directly below of this one is enabled to select a font out of it; all these fonts are in no way optimised for output via a laser scanner system, their intended target is publishing, therefore their lines and paths are optimised for quality which may result in a slow output

Laser Vector: this special type of fonts is optimised for laser marking operations. They consist of as less as possible lines and are designed for single-lined outputs like they are done with lasers. Using these fonts marking of texts is much faster than with the TrueType fonts. When this font type is selected the font list below of this combo box is modified to show the available laser fonts to choose from.

- Standard outline – this option is enabled only in case TrueType font type is selected and it is set by default. When a normal TrueType font is used, it always should be set. This option becomes useful when a non-standard TrueType font is used that is designed as laser font and – different to what TrueType fonts are designed normally – consists of single strokes only. Typically the characters of a TrueType font consist of an outline that forms the shape of the character and can be filled with a hatch pattern. Due to the lots of lines such a font is slow for laser marking. A faster, speed-optimised font would consist of single lines only which itself define the shape and typically do not need to be filled. Now when this option “Standard outline” is unset, such special fonts can be used (as TrueType does not know anything about laser marking, the option has to be reset explicitly by the user). Such fonts can be drawn in every standard TTF-editor and will look like laser fonts as soon as they are used in BeamConstruct and as soon as this option is disabled:



*From left to right: normal, outlined shape of a standard TrueType font, single line shape of a lasermarking-optimised font, view of a single line font when drawn into a TTF editor – the filled area is a wrong preview of most external editors and will be gone automatically as soon as this font is used in BeamConstruct and as soon as the option “Standard outline” is disabled there*

- Font – this list of available fonts can be used to select a specific font and style for the text. Depending on the font type chosen above, here a specific font can be selected. In TrueType mode the number and types of available fonts depends on the operation system and which ones are installed there. In Laser Vector mode the internal laser fonts are listed here, these fonts are delivered with BeamConstruct and do not depend on the fonts installed on the system.
- Alignment – the text is always drawn from a fixed position that is specified by the left mouse click when setting the text initially; using this list it can be chosen how the text has to be aligned to this position (left, centred or right)
- Orientation – this list specifies the direction the single characters of the text have to be drawn
- Spacing – this value can be used to modify the distance between the characters; here a value of 100% is the distance that is specified by the used font, smaller values set the characters closer to each other, larger ones stretch it to a bigger distance
- Kerning – this value can be used to influence the kerning distance for fonts that support it, here 100% is equal to the kerning value that it specified by the font, bigger values set the characters with a kerning value closer to each other, values smaller than 100% set a bigger distance between them; this parameter is different to the “Spacing”-value, it only influences character-combinations where kerning is defined for
- Arc Length – when creating geometry out of the available font information depending on the style of the chosen font different round parts have to be created (like the two arcs that are needed for creating a letter “B”). This parameter specifies the segment length for these round parts, the smaller it is the more accurate the result is but the more data are used to hold it.



- Nominal height – here the nominal height of a line of text can be specified in unit mm; this does not only include all possible characters and their heights but also the distance between two lines of the chosen TrueType font. So the height specified here in most cases will be always bigger than the vector height of every possible characters of the chosen font.  
Beside of that this is the nominal value on creation of the text, means no additional scaling operations which are done with this element or with optional parent groups elements are included in this value.
- Set monospaced – when this option is checked, equal distance is set between all characters independent from their shape and independent from the distance information of the used font
- Lock size – this option can be used to force a text element to always keep a specific size. When this option is enabled, the current geometric size of the element is stored. Whenever the text changes – either by manual modification, by remote access via a command interface or because the related input element changes the data – the element is scaled in a way so that the size is left unchanged. Depending on how much the real length of the texts changes, this means the characters itself are shrunk or stretched

To create radial text this element can be combined with the Curve Post-Processing Tool (please refer below)

#### 5.8.4.11 **Barcode**

Using this function several types of 1D and 2D barcodes can be created. This element accepts static data for creating a barcode or dynamic values that can be set via an Input Element. The size of the created barcode can be defined freely, when creating this element a rectangle is drawn exactly in the same way like it is done for the Rectangle Primary Element. This rectangle describes the outline of the barcode. For some barcode types that use a fixed aspect ratio between width and height the predefined outline is modified automatically.

The complete layout and structure of the barcode is defined via the parameters that can be set within the elements tab-pane:

- Data – the text or numeric data that are encoded within the barcode, here it depends on the chosen barcode type if and which characters, numbers or combinations of them are allowed and will result in a valid barcode. This means it is not possible to enter any kind of data for any type of barcode! For most barcode types special formatting rules need to be respected, elsewhere no barcode is created and the primary element stays empty.  
For these barcode types that support carriage return and/or line feeds (“\r” and/or “\n”), the related characters can be added by setting some special placeholders. So a “[CR]” within the data will be replaced by a carriage return character before the barcode is encoded and a “[LF]” will be replaced by a line feed character.
- Type – here the type of barcode can be chosen, depending on this type several of the following options are available and can be used for further definitions that are necessary for this type
- Mode – this is a barcode-type-specific option, here an encoding mode can be chosen for some of the barcodes
- Error Correction – this is a barcode-type-specific option, here an error-correction level can be set
- Size – this is a barcode-type-specific option, here the resulting size can be specified (which not influences the dimension of the barcode but the encoded data size)
- Bar Width – this parameter influences 1D barcodes only, here the size of the resulting bar lines can be modified; this value might be changed depending on the method of material processing in order to get a readable barcode
- Segments – this parameter is used only for the Maxicode barcode, it specifies the number of segments and therefore the smoothness of its circles
- Fade In Size – this parameter is used for inverted barcodes, it defines the size of the additional frame that is drawn around the barcode to get an inverted representation
- Force to Square – this option applies to DataMatrix barcodes only, in case it is set, the resulting barcode is always a square and never a rectangle

- Merge Cells – several barcode types may consist of cells (rectangles or squares) that are located directly beside each other; this option merges such cells wherever that is possible without destroying the barcode information, it can be used to save vector data
- Draw Cells as Dots – when this option is enabled, the cells some barcode types consist of are replaced by plain dots
- Draw Cells as Custom Dots – when this option is enabled, the cells some barcode types consist of are replaced by the custom dot geometry as defined within the settings (please refer to section 5.5 Project Configuration above)
- Invert – barcodes created by this element consist only of the outlines, together with the Hatching Additional Geometry element these outlines can be filled. Depending on the used material processing method these filled areas may result in lighter or darker areas. In case the result is not readable by barcode scanners the light and dark areas can be exchanged by this option, here an additional border is drawn around the barcode so that the hatching function generates an inverted fill pattern
- Invert with embedded normal – this option combines the normal mode and mode “Invert” into one element. When this mode is enabled, the visual representation of the barcode is switched to “inverted” with the possibility to define fade in size for the additional bounds created around it. This inverted barcode will make use of the pen assigned to the element as usual. Additional (but not directly visible) a non-inverted, normal barcode is created too making use of exactly the same input data like the inverted one. This non-inverted barcode is located at exactly the same position overlapping its inverted pendant perfectly. When this mode is enabled, the pen to be used for the embedded, non-inverted (=normal) barcode can be selected using the pen combobox which is shown right below the checkbox of this mode.  
This special mode can be used e.g. in cases where both, inverted and normal barcode needs to be marked with different pen parameters in order to have a good contrast on resulting material. With this mode it is not necessary to manually overlap two separate barcodes and it is guaranteed they both have the correct position and the same input data.

#### **5.8.4.12 Delay**

This element does not have a visual representation, it does not accept any additional data or post-processing but it can be used to influence the material processing. The elements of a project are processed in the order they appear within the Element Tree. At the position where this element is added, the full process is delayed for the given time. Thus this element can be used whenever a delay is necessary e.g. to let material become dry, let material become colder or whatever.

This element expects exactly one parameter:

- Delay – the delay (in unit seconds) the full process has to stop

PLEASE NOTE: the accuracy of the delay given here depends on the used hardware. When the scanner controller supports delays, it is handled by it directly and is normally very exact. When the scanner controller does not support this, the delay is done within BeamConstruct and much less accurate since timer resolution of the host operating system is quite low and transmission times to and from scanner controller card are unpredictable.

#### **5.8.4.13 Motion**

This element does not have a visual representation, it does not accept any additional data or post-processing elements but it can be used to influence the material processing in general. The elements of a project are processed in the order they appear within the entity tree. At the position where this element is placed additional motion axes (apart from the axes of the used scanner) can be driven.

The motion can be influenced by the parameters that have to be set for this element. First of all up to three axes can be controlled by the element using following parameters:

- Enable – specifies if an axis has to be changed by the following parameters or if it has to be left untouched, axes that are not enabled stay at their current position
- Position – the new position to move the axis to; this field is highlighted in yellow when the current settings of the axis may lead to a situation where the position value entered here is not used but overwritten during marking. This is possible e.g. in cases where an axis is bound to the height of a slice while marking 3D slice groups. For additional information please refer to axis settings as described in section “5.5 Project Configuration”.
- Speed – the speed for the movement to the given position
- Relative – defines a non-absolute movement, means the axis does not drive to the given position but changes its current position by the given distance
- Stop axis – the motion of an axis is stopped immediately, this option is available only for motion controllers that support this function and ignores the given position and speed values
- Move axes to home-position – the axis is moved to its home / reference position, this option is available only for motion controllers that support this function and ignores the given position and speed values
- Don't wait for end of motion – in normal operation mode full processing is halted as long as a motion operation is running. But when this option is set, this is no longer the case: then the motion operation is started and without waiting for its end, the following elements of the current project are executed. For such an asynchronous motion operation some things have to be noted:
  1. Prior to any other motion operation to be started, the current one needs to be stopped by adding another Motion element with the “Stop axis” option set.
  2. At the end of the current project it is recommended to add another Motion element with the “Stop axis” option set in order to end the project with a defined state
  3. This option is always valid for all axes of a motion controller, it is not possible to wait for the end of one axis and to let another axis work asynchronously

All other parameters influence the whole motion element:

- Pre-Motion Delay – this delay is done before the motion is triggered
- Use Control Output – the “BeamConstruct to Control”-plug-in provides two CTRL-outputs where motion control data can be emitted, this option gives the possibility to define which of both outputs OUT2 or OUT3 have to emit the motion information defined for this element

#### **5.8.4.14 Custom Output**

Most scanner controller cards provide several analogue and/or digital outputs. As soon as a scanner controller card is configured within the project settings this element can be used to send values to these outputs. This Primary Element does not contain any geometry so clicking the related tool-bar button immediately adds it to the current project.

The exact layout of the related property panel highly depends on the capabilities of the used scanner controller card. There options are shown depending on the outputs the used card provides.

So it is possible that following types of outputs can be enabled for modification and set within this element:

- Laserport – a numerical value according to the valid range of the (LP8) laserport (e.g. 0..255) can be entered, this value is sent to the card during processing in case the related check box is enabled
- Pilot laser – a digital output which normally is used to turn on and off the pilot laser automatically and therefore is assigned to the laser functionality directly. When such an output interface is available and not used anywhere else by the current configuration, it can be set to HIGH (“1”) or LOW (“0”)
- LaserB – a digital output which normally is used to control the laser timing automatically and therefore is assigned to the laser functionality directly. When such an output interface is available and not used anywhere else by the current configuration and configured to be used as general purpose output, it can be set to HIGH (“1”) or LOW (“0”)

- Main oscillator – a digital output which normally is used to control the laser automatically and therefore is assigned to the laser functionality directly. When such an output interface is available and not used anywhere else by the current configuration, it can be set to HIGH (“1”) or LOW (“0”)
- Digital – several buttons are shown that influence the related output ports bits and can be toggled into different states: “X” means this bit is left unchanged, “1” sets a bit to HIGH and “0” clears it; here the first button is the highest bit while the last button is always bit 0
- Analogue – a numerical value according to the resolution of the analogue output (e.g. 0..1023) can be entered, this value is sent to the card during processing in case the related check box is enabled
- Pulse Output – when this option is not set the output(s) simply is/are set according to the settings that are done above. In case this option is enabled and a time in unit milliseconds is given, the output is switched to the defined state only for this time, after that it is set back to its previous state. So this option can be used to emit only pulsed signals instead of a permanent output value.

PLEASE NOTE: The scanner controller card has to be set up and configured BEFORE this element is used for the first time. This is necessary because the capabilities of the scanner controller card need to be known for this element so that it can show the correct output options that belong to the card. When a completely different scanner controller card is selected for a project, it is necessary to save this project, close and restart the application to force a re-initialisation, load the project again and then check if all Custom Output elements still use the correct ports and values.

This element is symbolised by this toolbar icon:



#### 5.8.4.15 Custom Input

Most scanner controller cards provide several digital inputs. As soon as a scanner controller card is configured within the project settings, this element can be used to read values from these inputs and to halt processing of current project until a predefined state is found there. This Primary Element does not contain any geometry so clicking the related tool-bar button immediately adds it to the current project.

The exact layout of the related property panel highly depends on the capabilities of the used scanner controller card. There options are shown depending on the outputs the used card provides.

So it is possible that following types of inputs can be enabled and used within this element:

- Digital – several buttons are shown that read the related input ports bits and can be toggled into different states: “X” means this bit is ignored, “1” expects a HIGH value at this input and “0” a LOW; here the first button is the highest bit while the last button is always bit 0
- Serial interface – when supported by the used scanner controller card, data can be read from the serial interface periodically by this element. A comparison is done with the data read from this serial interface, here it can be selected if this has to be done with a full string (complete match) or with a substring (partial match).
- Display message during wait – when this option is set, a text can be entered which then is shown during operation as long as the software is waiting for the input signal to arrive as specified by the options above. This text is shown in a top-level window which is closed only when mark operation is stopped or when the expected input signal is detected and processing of the current project continues
- Timeout – when enabled, here a time (in unit seconds) can be specified after which the current operation will be cancelled also when the signal/data the plug-in waits for, has not arrived in the way as specified. When the timeout-condition has been met, the program flow continues in one of the ways specified by the listbox below of this option:
  - Continue marking – the operation continues in the same way as it would when the expected data would have arrived
  - Restart project – the marking cycle starts from the very beginning, means the project will be repeated
  - Stop marking – no matter what appears after the Custom Input element, when the timeout happens, the marking operation is cancelled. When, at the same time, in mark-dialogue the option “repeat” is set, the project is restarted
  - Go to element – the project will be continued at the position of the chosen element

This element is symbolised by this toolbar icon:



#### 5.8.4.16 Program Flow

While the Custom Input element can define conditional jumps (together with the timeout-function and depending on this timeout), this element can be used to perform non-conditional jumps within the current project. This typically makes sense only together with the conditional timeout-jumps of the Custom Input element and gives the possibility to define different sections within a project that have to be used exclusively dependent on this condition.

This element offers the following possibilities to jump to:

- Restart project – the marking cycle starts from the very beginning, means the project will be repeated
- Stop marking – no matter what appears after the Custom Input element, the marking operation is cancelled. When, at the same time, in mark-dialogue the option “repeat” is set, the project is restarted
- Go to element – the project will be continued at the position of the chosen element

This element is symbolised by this toolbar icon:



#### 5.8.4.17 External Trigger

This element does not contain geometry. After clicking the related tool-bar symbol it is added to the project immediately. Whenever the project reaches this elements position during execution, all output is stopped until an external trigger signal is detected. Here the external trigger according to the used scanner controller cards capability is used. Within the elements tab-pane the controller card has to be chosen, where the external trigger will appear.

For some controller cards, it is also possible to release such a trigger automatically when a given distance has been passed by (in marking on-the-fly applications), then the application does not wait for an external start signal. When this option is chosen by selecting the related checkbox “MOTF-Distance”, an encoder signal needs to be available and need to count into the correct direction, elsewhere the project will wait endless at this position.

This element also accepts an input element such as the CSV Data Input. When such an input element is configured, given input data are used to set a new MOTF-distance. The input data need to be whole-numbered values in unit micrometers. When a value equal or smaller than 0 is given or when invalid data (such as some texts) are given by the input element, the MOTF-distance is not modified and stays unchanged.

PLEASE NOTE: starting the geometry after a trigger is received is done by the scanner controller, the time between detection of the external trigger signal and output of the next geometry depends on the used scanner card and typically is less than some microseconds.

Waiting for a trigger after the last geometry has been sent (or after the last Primary Element was processed completely) is controlled by the application. Here the reaction time highly depends on the used operating system and the available computing power. So it is recommended to ensure a gap of at least 20 milliseconds after the last element was processed until the external trigger may appear.

This element is symbolised by this toolbar icon:



#### 5.8.4.18 Serial port output

This element does not contain any geometry but can be used to output some data to a serial port connected to the host PC where the application is running at. To submit data via the serial interface integrated into a connected scanner controller, please refer to section “5.8.4.14 Custom Output” above.

Within the element tab-pane the serial port has to be configured which has to be used for data transmission. This includes the port name, transmission rate and port parameters.

Below of the serial port configuration elements, a text field “Data” is available where the data have to be entered which need to be submitted. Here all types of ASCII-texts can be given as well as the two special place-holders [CR] and [LF] for adding a carriage return (“\r”) and a line feed (“\n”) to the data. A line break done in the text field is not converted to such characters but dropped, so it is mandatory to use these place-holders at all positions where this is required.

This element also accepts an input element such as the CSV Data Input. When such an input element is configured, given input data are used to set new data to be sent via the serial port. These input data need to follow the same rules as described above in order to add correct line feed/carriage return characters to the data stream.

Sending of the data is done synchronously, means when such an element is found, application waits until processing of data is finished for all scanner controller cards. Then the data are sent but without waiting for a response before processing the next element in current project.

This element is symbolised by this toolbar icon:



#### 5.8.4.19 Z-Shifter

This element does not have a visual representation but is similar to the motion primary element. It does not accept any additional data or post-processing elements but it can be used to influence the material processing in general by changing the lasers focus. The elements of a project are processed in the order they appear within the entity tree. At the position where this element is placed, a focus shifter or beam expander is accessed. The related Z-Shifter plug-in has to be configured in projects hardware settings.

The focus/z-position can be influenced by the parameters that have to be set for this element:

- Position – the new position to move the z-position to
- Speed – the speed for the movement to the given position
- Relative – defines a non-absolute movement, means the axis does not drive to the given position but changes its current position by the given distance
- Move axes to home-position – the axis is moved to its home / reference position, this option is available only for motion controllers that support this function and ignores the given position and speed values

All other parameters influence the whole motion element:

- Pre-Motion Delay – this delay is done before the motion is triggered

### 5.8.5 Additional Geometry Elements

Additional Geometry elements add new geometry based on the already available geometry of the primary element it belongs to. Here the additional element only takes care of the primary one, it ignores possibly existing, other additional elements that might exist for the same primary element. Additional Geometry elements do not modify any of the existing geometries but only add own ones to them.

To add an Additional Geometry element to an existing Primary Geometry, the drawing area doesn't have to be used. Here the desired primary element has to be selected within the element tree, then the additional element has to be clicked within the tool-bar. There the Additional Geometry elements are shown in purple colour and are added to the Primary Geometry element as soon as the tool-bar item is clicked.

### 5.8.5.1 Hatcher

Hatching is the process of filling a closed polygon with a pattern. This can be used to process a bigger area. It can be useful e.g. combined with texts to get characters that consist not only of its outline.

Hatching will work only with closed polygons and might fail on specific geometries that do not describe a closed area non-ambiguous. In such cases the resulting hatch pattern may be incomplete or may be located outside of the polygon.

In general only two-dimensional geometry can be hatched. Lines and dots – which do not describe a closed polygon – are not extended by this element. As an other condition a polygon can be hatched only at positions where the outline has a closed line style, geometry with dotted lines can't be hatched, geometry with a dashed outline is hatched only at positions where a line is drawn.

Also in case 3D mode or 3D geometry is used the hatcher offers a 2D-functionality only but it can be applied to 3D elements. In such cases the hatcher always uses the geometry in X and Y direction (similar to the top view) and ignores the Z-value to create the hatch pattern. After this pattern is generated the Z-information out of the lines they start from/end at are used to modify the hatch lines. This results in perfect 3D hatch patterns in most cases but may have unexpected results when complex geometries are combined with more advanced hatch options (like beam compensation or inner hatch lines).

The hatcher offers following parameters:

- Line Style – the style the lines the text consists of is created with, here following options exist:
  - Continuous: the geometry consists of closed, connected lines with as less as possible points
  - Connected Lines: the geometry consists of several short lines including additional points, all these lines are connected; from its visual representation this looks identically to “Continuous” but behaves different when a post-processing element is applied to this hatch afterwards
  - Dashed (simple): the geometry consists of several short lines including additional points, every second line is drawn that results in a dashed line style; this mode starts with a line at every beginning of a hatch line
  - Dashed (interleaved): the geometry consists of several short lines including additional points, every second line is drawn that results in a dashed line style; this mode starts with a line and a space alternative at every new beginning of a hatch line
  - Dashed (continuous): the geometry consists of several short lines including additional points, every second line is drawn that results in a dashed line style; this mode starts with a line or a space at the next beginning of a hatch line depending on the previous hatch line, the pattern is kept continuous
  - Dotted: no lines are drawn, the geometry consists of a set of not connected dots
  - Dotted (Custom): no lines are drawn, the geometry consists of a set of not connected points in custom dot style as configured within the project settings (please refer to section 5.5 Project Configuration above)
  - Dotted pairs: no lines are drawn, the geometry consists of a set of not connected pairs of dots with the given distance between each pair and half of the distance between the dots of a pair
  - Dotted pairs (custom): no lines are drawn, the geometry consists of a set of not connected pairs of dots with the given distance between each pair and half of the distance between the dots of a pair using the custom dot style as configured within the project settings (please refer to section 5.5 Project Configuration above)
- Distance – whenever a non-continuous line style is selected above, this distance value specifies the length of the additional points that form the selected line style
- Offset – whenever a non-continuous line style is selected, the related pattern is calculated in a way so that it starts at the beginning of the related hatch line. Dependent on the filled shape this may result in line patterns which are aligned in a way which is not intended. Using this parameter an additional offset can be added to each line incrementally, so that the beginning of the hatch pattern is shifted by this offset. As soon as the real, accumulated offset value becomes bigger than the line distance defined with previous parameter, the distance is subtracted from the current offset value
- Align dots to raster – whenever a non-continuous line style is selected, the related pattern is calculated in a way so that it starts relative to the beginning of the related hatch line taking an optionally set offset-value into account. When the current hatch angle (as described below) defines an exact horizontal or vertical line (0, 90, 180 or 270 degrees) and this option is set, the pattern for

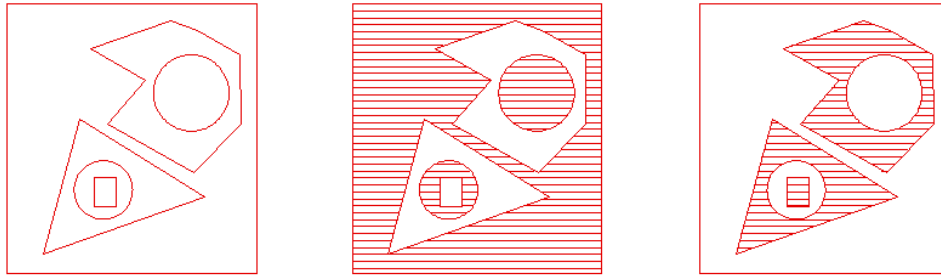
the line is no longer calculated relative to the beginning of the line but aligned to the coordinate 0,0 of the underlying coordinate system. This is useful e.g. for polygons with different edge angles where e.g. a dot-pattern would result in an irregular, uneven distribution of dots.

- Hatch Style – here the hatching mode is defined, depending on the selection made here a different pattern is used to fill a polygon
  - Lines, forward – filling is done with parallel lines that all go into the same direction
  - Lines, forward/backward – the polygon is filled with parallel lines that change their direction
  - Connected lines – this mode extends the previous one, here the parallel lines are connected and form some kind of waved lines
  - Zig-zag – these lines are no longer parallel, they exchange their direction alternative and are connected at their ends directly
  - Inner lines – this mode repeats the outline of the hatched polygon in its inner side as often as it fits with the chosen hatch distance
  - Outer lines straight – this mode is similar to "Inner lines" but not a plain hatch mode that can be used to really *fill* a geometry. It repeats the outline of the hatched polygon on its outer side as often as specified by the value given in field "Number of Lines". At corner points where additional lines are necessary to close the polygon, the shortest way is used by adding a straight line.
  - Outer lines rounded – this mode is similar to "Inner lines" but not a plain hatch mode that can be used to really *fill* a geometry. It repeats the outline of the hatched polygon on its outer side as often as specified by the value given in field "Number of Lines". At corner points where additional lines are necessary to close the polygon, a round shape is created. Please note: this adds a lot of polygon points and may – dependent on the geometry – increase memory and processing time consumption dramatically. So this mode should be handled with care and only in cases where it is really necessary to use this special shape!
- Hatch Angle – this angle specifies the orientation of the hatch lines according to the coordinate system, it does not apply to the inner/outer lines hatch modes
- Angle Offset – this parameter applies only in case the hatcher is set as sub-element of a group that contains more than one geometric element to be hatched; in this case only the first primary element within that group is hatched with the original hatch angle, for all following elements the hatch angle is modified by this offset. So this parameter can be used to have some kind of rotating hatch angle for several elements that are concatenated together within a group. It does not apply to the inner/outer lines hatch modes.
- Repeat – this option becomes enabled when an angle offset is defined with previous parameter. When "Repeat" is set to values greater than 0, the hatch generation is repeated for the given number, each with a different angle according to the given offset. So this function can be used to create complex hatch patterns with rotating angles while using one hatch element only.
- Hatch Distance – this is the distance that is kept between the hatching lines, for the zig-zag hatching mode it applies to the ends of the lines
- Hatch Offset – normally hatching starts at the border of a polygon; this parameter allows to specify an offset (in positive and negative direction) to modify the position where hatching really starts
- Align hatch to raster – when the current hatch angle (as described above) defines an exact horizontal or vertical line (0, 90, 180 or 270 degrees) and this option is set, the hatch lines are no longer calculated relative to the beginning of the element to be hatched but aligned to the coordinate 0,0 of the underlying coordinate system. This is useful e.g. when different independent elements have to be hatched with the same hatch lines alignment.
- Number of Lines – this input field is activated only when "lines" hatch styles are used; it specifies the maximum number of inner or outer lines to be calculated; when this value is smaller than the number of inner lines required to fill a geometry completely, inner part of the element will contain a non-hatched hole
- Interleave Lines – when this parameter comes with a value greater than 1, hatch lines are not marked one after an other but by dropping that number of lines after each, that is specified by this parameter. The remaining lines then are marked in additional cycles skipping the same number of lines in between. This mode slows down marking process caused by many additional jumps but gives the possibility to let the material not heat up too much. This can be useful in case of temperature-sensitive materials, there the currently marked areas get some time to cool down until the lines in their direct neighbourhood are processed. As an example: instead of marking lines 1, 2,



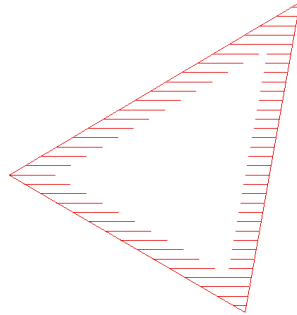
3, 4, 5, 6, 7, 8, 9, 10, 11, 12 with an interleave-value of 4 it would mark lines 1, 5, 9, 2, 6, 10, 3, 7, 11, 4, 8, 12.

- Invert – this function applies to nested polygons only: when it is set, the order of hatched/not hatched polygons is inverted as shown in picture below:

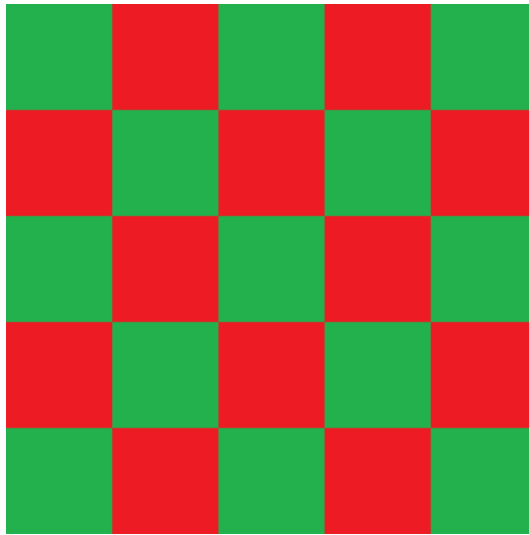


From left to right: input geometry with nested polygons, hatched geometry (no-inverted), inverted hatched geometry

- From bounding box – hatching is started from the (not visible) bounding box, when this option is set, hatch lines start outside of the polygon; this function should not be used in cases where the bounding box is the same as the shape of the hatched geometries which is true for all rectangles which are aligned parallel to the working area border
- At border – when this option is set, polygons are no longer filled completely but only at the borders, the given distance in unit specifies the “thickness” of the resulting border hatch in unit mm:

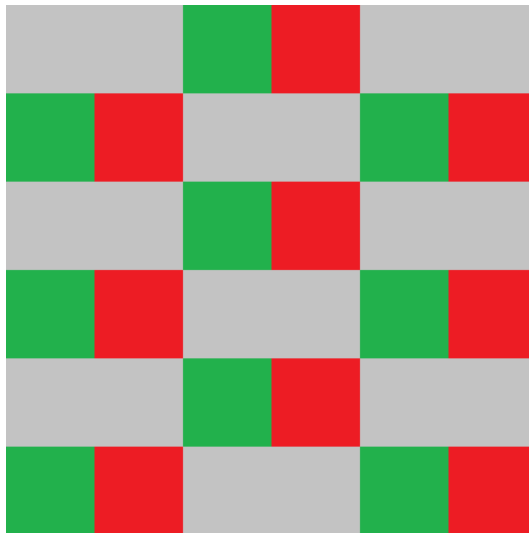


- Beam Compensation – the laser beam that is used to process a working piece itself has an own size that results in the width of the lines that are processed by it; for the hatcher it might be necessary to compensate this lasers spot size to avoid parts of the working piece to be processed more than once by the overlapping beam, this compensation can be chosen here:  
Use spotsize from pen – here the tool compensation sizes are calculated automatically out of the spot sizes of the used pens  
Custom –when this option is used a custom value for compensation can be entered
- Split To Tiles – when this option is set, the hatch lines are split into tiles that are aligned to the working area and that subdivide the hatch pattern additionally. Using these tiles it can be avoided to heat up the material during hatching too much. When this option is set, the following parameters can be used to specify exact behaviour of this function. After enabling this option, the hatches are split and reorganised but there will be no obvious visible representation of the changes, it can be noticed during marking or via the coordinate values only.  
Together with the following parameters this function can be used to e.g. mark every second tile:



Assumed the working area has a size of 100 mm x 100 mm, for this behaviour "Tile Size" needs to be set to 20 mm and "Tile Skip Size" has to be set to 2. With these settings first all these areas are marked that have vectors within red tiles, finally all the remaining areas that have a vector in green tiles are marked. Tiles itself are processed from upper left to lower right corner row by row.

It is also possible to have even bigger distances between tiles so that they do not touch at their edges:



Parameters for a 90 mm x 90 mm working area are 15 mm for "Tile Size" and 4 for "Tile Skip Size". This results in a behaviour where marking starts with these areas of the hatch that own vectors in red tiles, next the areas in green tiles are marked. This is continued until the whole working area was processed (in this example by processing the remaining two tile sets in grey areas with two additional steps).

- Tile Size – the size of a tile in working area, this value should be such a fraction of the working area size that lets the tiles fit fully into whole working area
- Tile Overlap – when this parameter is set to a value greater than 0, it specifies how much neighbouring lies should overlap, when it is smaller than 0, the value sets a gap between such neighbouring tiles using the given size in unit mm
- Tile Skip Size – skip distance between tiles, this value decides how many tiles in horizontal direction should be skipped before marking continues in next tile; remaining tiles are processed on next cycle; this value should be much smaller than the total number of tiles that fit into the working area
- Tile drop offset – when this parameter is set to a value greater than 0, the hatcher no longer creates a complete set of tiles resulting in a closed hatch for the complete element. Instead of this, the

parameter specifies at which tile the function should start dropping the tiles so that there exist non-hatched parts in the whole geometry. This function can be used to combine several hatches with different hatch angles in tiles. As an example: when one hatch is set with a hatching angle of 0 degrees and a tile drop offset of 1 and the next hatch on the same element is set with a hatch angle of 90 degrees and a tile drop offset of 2 (while both use a tile skip size of 2), the result is a closed, tiled hatch with alternating hatch angles of 0 and 90 degrees in the different tiles.

- Hide hatch from view – this option does not influence generation of data, so no matter if it is set or not, the marking result will stay the same. When this checkbox is set, the hatch is not shown in the drawing area in order to make displaying of the whole project faster. This can be useful in case of very complex hatch patterns with millions of single lines or dots
- Save as default – when this button is pressed, the current hatch parameters are saved as default values. On next application start and on first usage of the hatch element, these saved parameters then will be used instead of the predefined ones

## 5.8.6 Post-processing Tool Elements

The Post-Processing Tools influence all geometric elements that are located on an earlier position within the Element Tree and below of the same primary element. So the manipulation of geometry that is done by such a tool is always applied to the Primary Geometry element it belongs to and to all previously applied Additional Geometry elements of this primary one. Additional elements that follow a tool are not changed, they apply their additional geometry to the modified geometry.

Post-Processing Tools are symbolised by red tool-bar buttons and can be added to a primary element simply by clicking this tool-bar button. To do that, the related Primary Geometry element has to be selected within the Element Tree.

Post-Processing Tools do not add any new points or lines to existing geometry, thus it requires an outline that consists of lines that are short enough to let the tool apply its function, elsewhere the result would not look like expected. So within the parameters of the elements that have to be modified the line-style has to be set to a non-continuous mode (for an example connected lines, dashed lines or dots with a small distance should be used here).

### 5.8.6.1 Sine Distortion

This Post-processing Tool applies a sine waveform to existing geometry. This tool does not add any new points or lines to existing geometry, thus it requires an outline that consists of lines that are much shorter than the wavelength. Elsewhere the result would not look like a real sine waveform.

Following parameters can be set for this tool:

- Direction – the direction the waveform has to be applied to, beside the two values that are relevant for 2D data ("Horizontal" sets a sine parallel to X-axis, "Vertical" sets a waveform parallel to Y-axis) four additional options are available that are relevant for 3D data only. These options can be used to specify the direction of the amplitude and the direction the wave grows into
- Wavelength – the length of a single wave
- Amplitude – the amplitude of the sine wave
- Offset – an offset that defines at which position within the wave the sine calculation has to be started, this offset can be used to shift the waveform

### 5.8.6.2 Curve Distortion

Using this tool an existing geometry can be bend around a defined centre point to get a radial, round shape. It is useful e.g. in combination with the Text Primary Geometry element to define radial or ellipsoidal texts.

This tool does not add any new points or lines to existing geometry, thus it requires an outline that consists of lines that are short enough for a result that looks like a bent outline.

Following parameters can be used to influence the result:

- Axis – this field specifies around which coordinate axis the text has to be bent. Here “Z” is preselected and can be used for 2D applications. “X” and “Y” are for 3D applications and bend the post-processed geometries in Z-direction.
- Radius – the radius of the circle the geometry is bent to, this is the distance from the middle of the geometry to the radial centre point; here negative values are allowed too to invert the geometries position relative to the chosen axis
- Radius – the second radius value can optionally be enabled in order to create ellipsoidal shapes. In this case a different radius can be entered specifying the ellipses second radius. Please note: when the first radius is negative, the second one has to be negative too, elsewhere the results are undefined
- Angle Offset – an offset for the bending angle starting point, using this value the geometry can be shifted
- Force to centre coordinate – when bending some geometry, this is done using some (imaginary) coordinate which is used as centre for the geometry. When this option is set, the centre coordinate can be entered directly within the settings of this plug-in, forcing the geometry to a position with exactly this centre coordinate. This makes it easier to properly adjust and position a bent element. PLEASE NOTE: when this option is set, all other functions to translate or change the position of this element are disabled in order to avoid overlapping coordinate transformations! So as long as this function is used, the element can be repositioned only via the X and Y coordinate input fields of this post-processing element!
- Rotate CCW – rotate the parent-element counter-clockwise. This mainly is useful in case of text-elements used for rotation.

## 5.8.7 Data Input Elements

This type of element influences geometry only in a very indirect way, it provides data that can be used by Primary Geometry elements to create their geometry. So all types of Primary Geometry elements that accept such input data can be combined with all types of Data Input elements. Of course it is not possible to have more than one Data Input element per Primary Geometry element.

To add such a Data Input element, the Primary Geometry element has to be selected within the Entity Tree. Next the green tool-bar symbol of the Data Input has to be clicked to assign it to the Primary Geometry element.

### 5.8.7.1 CSV Data Input

This plug-in offers connectivity to data out of tables in CSV format (comma separated values, a platform- and application-independent table file format). It loads data out of such a CSV table and jumps to next line/row of this file automatically during marking. The data read from this table are applied to the related Primary Element. Within the settings panel of this plug-in first the path to the CSV file has to be chosen, then the related input elements become enabled which can be used to configure the behaviour of this element:

- Column separator – here the delimiter character has to be chosen which separates the data columns within the selected CSV file
- Data column – using this field the column has to be specified where the data have to be read from, this value is 1-based
- Data repeat column – using this checkbox and input field a function can be activated where an other column in the same CSV file specifies how often the data from the previous configuration field have to be repeated until the next row of data has to be read out of the file. The repeat-value given here is

0-based, means with a value of 1 data out of the current row would appear two times (once the original value and once repeated due to the repeat column value)

- Endless loop – when this checkbox is set, the application will jump to the beginning of the CSV file as soon as all data have been read. When this option is not set and when there are no more data available within the CSV file, a marking process will be cancelled with a related error message.

During marking in Element Tree the name of this element is prepended by two numbers showing the current row of the loaded CSV file and the maximum number of rows it contains. Using these displayed values it can be seen how much CSV data of the current file are already processed and how much are still left.

### 5.8.7.2 *OpenAPC Data Input*

This input can't be previewed within the editor of BeamConstruct. It expects input data that can be provided e.g. via the BeamServer remote control (please refer to 5.21 for more details).

### 5.8.7.3 *Serial/Date/Time*

This is a very complex Data Input element that can be used to generate numbers, date or time information or combinations of them. The result can be used to create e.g. complex serial numbers, "best-before" information or others within a text or a bar code – depending on the Primary Geometry element where this input is assigned to.

The Data itself are updated before the next material processing cycle is started. Depending on the used configuration this update increments/decrements the serial number and/or updates the date or time information to represent the current time.

Following parameters can be used to influence the resulting data and their behaviour:

- Start Value – the starting value where the serial number has to count from when the project is loaded initially
- Reset At – here a number can be defined that is used as limit for the serial number, as soon as the current serial number is equal to the value given here the serial number is reset to the start value defined above
- Increment – the value the serial number has to be incremented by for every update; this value can be negative which results in a decrement for every cycle
- Beat Count – this number specifies the number of cycles a serial number has to be left unchanged until it is updated, here a value of three – as an example – would mean, a serial number is updated and incremented only at every third marking cycle
- Beat Offset – this parameter corresponds to the preceding one, it adds an offset to the beat count, means it defines at which number it has to be started to count for the number of Beat Counts
- Numeric Base – by default a serial number consists of numerical values in range 0..9 which means the numeric base is 10; this parameter can be used to define any other numeric base so that less numeral values are used (Numeric Base < 10) or in a way where letters in range A..Z are added for counting the serial number (Numeric Base > 10)
- Minimum Digits – this parameter specifies the number of digits a serial number has to consist of at minimum – independent from their real value; in case a number would have less digits than specified here, it would be filled with one or more preceding zeros
- Time Offset – here an offset can be specified using freely selectable numbers of years, months, days, hours and minutes, this offset is added to the time information that can be incorporated into the resulting data and can be used e.g. to define a best-before date counted from the current date/time
- Conditional offset – using this option, an offset can be added to the serial number depending on some condition. This condition can be described using the input elements of this parameter section: "if <value> is <comparator> than <compare value> add <offset value> to <serial number value>". The parts in brackets can be chosen freely to define a condition which makes use of a specific part

of the current serial number, compares it to a value and when this condition is met, a specific offset value is added to a specific part of the current serial number. Using this function, it is possible to define irregular and non-linear offsets and jumps within a serial number.

- Format String – this format string defines the layout and contents of the resulting data which are handed over to the assigned Primary Geometry element. Here any desired text can be added and combined with some place-holders which are replaced by serial number or date or time information. Here following placeholders are supported:
  - \$S – this string is replaced by the serial number as defined with the counting and count-reset parameters as described above
  - \$%ymd – this string is replaced by the current date in format YYMMDD; when any date- or time-offsets or conditional offsets are defined, they do NOT affect this date string, means it always uses the current local date
  - \$%dMY – this string is replaced by the current date in format DDMY where the month is represented by a character starting from "A" and the year is represented by a character starting from "A" for the year 2020
  - \$%HM – this string is replaced by the current time in format HHMM; when any date- or time-offsets or conditional offsets are defined, they do NOT affect this date string, means it always uses the current local time
  - %Z – is replaced by the time zone name
  - %z – is replaced by the time zone offset from UTC; a leading plus sign stands for east of UTC, a minus sign for west of UTC, hours and minutes follow with two digits each and no delimiter between them

Following additional placeholders can be used to construct own date and time representations; different to the predefined format identifiers above, when any date- or time-offsets or conditional offsets are defined, they are used by these format identifiers:

- %A – is replaced by national representation of the full weekday name
- %a – is replaced by national representation of the abbreviated weekday name
- %B – is replaced by national representation of the full month name
- %b and %h – are replaced by national representation of the abbreviated month name (these placeholders are not supported when using Windows)
- %C – is replaced by the year divided by 100 (this placeholder is not supported when using Windows)
- %c – is replaced by national representation of time and date
- %D – is equivalent to "%m/%d/%y" (this placeholder is not supported when using Windows)
- %d – is replaced by the day of the month as a decimal number (01..31)
- %e – is replaced by the day of month as a decimal number (1..31); single digits are preceded by a blank (this placeholder is not supported when using Windows)
- %F – is equivalent to "%Y-%m-%d" (this placeholder is not supported when using Windows)
- %G – is replaced by a year as a decimal number with century. This year is the one that contains the greater part of the week (Monday as the first day of the week)
- %g – is replaced by the same year as in "%G", but as a decimal number without century (00..99)
- %H – is replaced by the hour (24-hour clock) as a decimal number (00..23)
- %I – is replaced by the hour (12-hour clock) as a decimal number (01..12)
- %j – is replaced by the day of the year as a decimal number (001..366)
- %k – is replaced by the hour (24-hour clock) as a decimal number (0..23); single digits are preceded by a blank (this placeholder is not supported when using Windows)
- %l – is replaced by the hour (12-hour clock) as a decimal number (1..12); single digits are preceded by a blank
- %M – is replaced by the minute as a decimal number (00..59)
- %m – is replaced by the month as a decimal number (01..12)
- %p – is replaced by national representation of either "ante meridiem" ("AM") or "post meridiem" ("PM") as appropriate
- %R – is equivalent to "%H:%M" (this placeholder is not supported when using Windows)
- %r – is equivalent to "%I:%M:%S %p" (this placeholder is not supported when using Windows)
- %S – is replaced by the second as a decimal number (00..60)
- %s – is replaced by the number of seconds since January 1st, 1970, 00:00 o'clock, UTC (this placeholder is not supported when using Windows)
- %T – is equivalent to "%H:%M:%S" (this placeholder is not supported when using Windows)

- %U – is replaced by the week number of the year (Sunday as the first day of the week) as a decimal number (00..53)
- %u – is replaced by the weekday (Monday as the first day of the week) as a decimal number (1..7) (this placeholder is not supported when using Windows)
- %V – is replaced by the week number of the year (Monday as the first day of the week) as a decimal number (01..53). If the week containing January 1st has four or more days in the new year, then it is week 1; otherwise it is the last week of the previous year, and the next week is week 1.
- %v – is equivalent to "%e-%b-%Y" (this placeholder is not supported when using Windows)
- %W – is replaced by the week number of the year (Monday as the first day of the week) as a decimal number (00..53)
- %w – is replaced by the weekday (Sunday as the first day of the week) as a decimal number (0..6)
- %X – is replaced by national representation of the time
- %x – is replaced by national representation of the date
- %Y – is replaced by the year with century as a decimal number
- %y – is replaced by the year without century as a decimal number (00..99)
- %+ – is replaced by national representation of the date and time (this placeholder is not supported when using Windows)
- %m – is replaced by the month representing it as a character in range a..j
- %M – is replaced by the month representing it as a character in range A..J
- %y – is replaced by the year representing it as a character in range a..z
- %Y – is replaced by the year representing it as a character in range A..Z
- Locale – some of the date/time place holders are replaced by information that depend on a language and or on local specialities; using this parameter the locale can be chosen that has to be used for these time information
- Letter set – some of the date/time placeholders are replaced not by numbers or texts but by single, consecutive letters out of an alphabet. Using this parameter this alphabet can be customised, so the letters given here are used instead of numbers/counting for these specific format placeholders.

## 5.9 Edit Geometry

Geometries are mainly generated by Primary Geometry elements. Whenever the parameters of such an element are changed the geometry is re-created and all assigned Additional and Post-Processing Elements are executed again to apply their modifications to this new geometry. Thus modifications that are done for single vectors get lost as soon as such modifications are done. Nevertheless it is possible to perform such modifications in case one of the two conditions are met:

- the parameters of the elements that create a geometry are not touched
- a geometry is converted to a static one where no Primary Geometry element with variable parameters is available and where no Additional Geometry or no Post-Processing Element is assigned

Functions to edit geometries can be found at two places: in menu “Edit” and its sub-menus “Non-Destructive” and “Destructive” within the main window and within the context menu of the drawing area that opens when an element is selected and the right mouse button is clicked.

### 5.9.1 Edit Vectors

Editing of one or more separate vectors requires selection of these vectors. This can be done within the Coordinates-list of the Geometry-tab-pane when an element is selected. Here following functions are available:

- Delete selected Points – all selected points are deleted, resulting from that the lines between them are removed too
- Delete selected Lines – all lines between the selected points are removed, the points itself are left untouched

- Insert point before – a new point is added in the middle between the currently selected point and its preceding one
- Insert point after – a new point is added in the middle between the currently selected point and its subsequent one

## 5.9.2 Edit Non-Destructive

The non-destructive editing functions perform modifications that leave the geometries and the elements they are created from untouched. Some of them are available also from within the context menu of the Element Tree. Alternatively they can be found in Menu “Edit” sub-menu “Non-Destructive” and require at least one element to be selected:

- Group – several selected Primary Geometry elements are concatenated together within a group
- Group to Active Split – using this function geometries that are larger than the available working area can be cut into smaller pieces and processed step by step either by horizontal or rotational movements using an additional drive; for more information please refer the related section below
- Group to Active Rotational Split – this is a convenience function which is similar to Active Splits mentioned above but which encapsulates the rotational part only
- Group to Active Move – here full geometries can be processed repeatedly and combined with movements so that the same geometry is arranged in tiles or at the circumference of a ring several times; for more information please refer the related section below
- UnGroup – an existing group of elements is removed, the contents of the group are put out of it
- Duplicate – a copy of the currently selected element is created, the copy is 100% identically to its original (except its position, it is placed a bit above the original element)
- Reverse direction – reorganises the vectors in a way where the geometry is kept unchanged completely, but with reversed marking direction; after processing an element using this function its marking orientation is changed and it starts marking where it previously has ended
- Align left – this function can be used to left-align several selected elements at each other. Here the left-most-position of all these elements is taken to place all elements at this position in horizontal direction
- Centre horizontal – this function can be used to centre several selected elements along each other. Here the centre-position of all these elements is taken to also centre-align all elements at this position in horizontal direction
- Align right – this function can be used to right-align several selected elements at each other. Here the right-most-position of all these elements is taken to place all elements at this position in horizontal direction
- Align top – this function can be used to top-align several selected elements at each other. Here the top-most-position of all these elements is taken to place all elements at this position in vertical direction
- Centre vertical – this function can be used to centre several selected elements along each other. Here the centre-position of all these elements is taken to also centre-align all elements at this position in vertical direction
- Align right – this function can be used to bottom-align several selected elements at each other. Here the lowest position of all these elements is taken to place all elements at this position in vertical direction

### 5.9.2.1 Active Split Group

This is a special kind of group that concatenates several elements into one group. Due to the kind of this group it is recommended to use only geometries and scanner bitmaps and not to put other ones like motion elements into it.



An Active Split Group is able to handle geometries and bitmaps that are larger than the available working space. It cuts them into smaller pieces and combines them with motion control information so that these large and cut geometries and bitmaps can be put together during processing. So this functionality can be used to handle big working pieces that e.g. are moved by a XY-table. The cutting operation only applies to X-Y-direction (equal to top view).

An other application is processing on the surface of rings, here a special radial mode is supported. This mode gives the possibility to cut geometries into segments and to process the whole ring (inner or outer side).

As soon as some geometric elements are grouped together within such an Active Split Group it's behaviour can be modified using the parameters that are available within the property panel of this group:

- Split horizontal – splits the grouped geometry along X axis, this option can be combined with “Split vertical” to define two-dimensional movements
- Split distance – the split size in horizontal direction
- Split vertical – splits the grouped geometry along Y axis, this option can be combined with “Split horizontal” to define two-dimensional movements
- Split distance – the split size in vertical direction
- Split radial – defines a splitting operation onto a ring, this option can not be combined with one of the previous splitting direction modes
- Direction – specifies the direction for the rotational splitting, here it can be chosen between “Horizontal” and “Vertical”, the axis that is assigned to this direction has to be aligned properly
- Diameter – the total diameter of the ring where the split pieces have to be marked at
- Split angle – the movement angle for every split, the size of the single split pieces result out of this angle and the diameter of the ring
- Keystone correction – when marking the inner side of a ring the laser beam has to be slanted in order to hit only the lower inner side of the ring and not the upper outer side too. Resulting from that the length of the beam varies dependent on the X/Y position of the geometry which results in a distortion when no special optic is used.  
Using the keystone correction this distortion can be suppressed so that the marking result will look like expected. This correction is done by setting an as a percentage correction factor and by defining the side of the geometry this factor may apply to. The correction factor specifies how much the geometry shall grow or shrink according to its original size at the outer position where the full distortion has to be expected.
- Invert motion direction – this option has to be used in case the inner side of a ring needs to be processed, there an inverted motion direction is required to concatenate the pieces in correct order
- Process order – this order applies to the planar splitting modes and defines in which order the pieces have to be processed:  
  - Left to right – horizontal direction always starts from left
  - Right to left – horizontal direction always starts from right
  - Zig-zag from left – horizontal direction starts from left and performs zig-zag movements
  - Zig-zag from right - horizontal direction starts from right and performs zig-zag movements
  - Top to bottom – vertical movement starts from top
  - Bottom to top – vertical movement starts from bottom
In case two-dimensional motion is defined both process order selections are combined with each other to describe the 2D motion.
- Motion speed: the movement speed in unit mm/sec or degrees/sec that is applied to the motion between the splits.
- Overlap splits: here an as a percentage value can be set that specifies how much the pieces should overlap, means how much of the geometry should exists twice so that it is processed two times at the areas of intersection
- Try split between pieces – here the splitting algorithm tries to find empty spaces between geometries to avoid cutting of lines; in case this option is set the specified split distance or split angle is used as maximum value: a cut is set wherever space between geometries is found or latest after the given

split distance/split angle. In last case cutting of lines is still done. So in this mode split distance/split angle define the absolute maximum size of a resulting piece while the distances where the geometry are cut really can be smaller than this.

- Axis mapping: at the lower end of the property panel a matrix of radio buttons exists where the axes can be mapped to the related movements, there it can be specified which of the axes A, B or C of the two possible outputs OUT2 and OUT3 have to provide the related planar movements in horizontal or vertical direction or the angular movement.
- Save as default – after application start-up new split-groups are created with some default parameters. When these parameters are changed, other Active Split Groups then are created with these changed parameters, but after restarting BeamConstruct, the default parameters are used again. When this button is pressed, the current split-group parameters are saved as new default so that they are used whenever a new Active Split Group is created after BeamConstruct was started newly. The related data are stored within the users working directory in a file named “.beamuiSPLT”

After such a splitting group has been created and after the correct parameters have been set, the appearance of the related geometry changes: now it is overlapped by horizontal and/or vertical lines. These lines specify the cutting positions and can be modified manually. They can be dragged with the mouse in horizontal or vertical direction exclusively to change the position where the geometry has to be cut. The related modifications are done as soon as such a cutting line is released.

Please note: the positions of the cutting lines do not influence the parameters given within the property panel but as soon as the parameters within the property panel are modified, the position of the cutting lines are reset to their default. Thus their positions should be changed only in case all other parameters do fit perfectly.

### **5.9.2.2 Active Move Group**

This is a special kind of group that concatenates several elements into one group. Due to the kind of this group it is recommended to use only geometries and not to put other ones like motion elements into it.

An Active Move Group is able to repeat a grouped geometry several times and to combine it with movements so that e.g. a ring can be processed with a repeated pattern on its surface or to repeat the current geometry on a large working piece that itself is re-positioned by using an XY-table. The active movement functionality only applies in 2D mode or in X-Y-direction (equal to top view). Comparing to the Active Split Group here the contents of such a group are not cut or somehow else modified, they are marked all in once repeatedly.

As soon as some geometric elements are grouped together within such an Active Move Group the groups behaviour can be modified using the parameters that are available within the property panel of this group:

- Move horizontal – moves the grouped geometry along X axis, this option can be combined with “Move vertical” to define two-dimensional movements
- Move distance – the distance a movement has to be done in horizontal direction between two processing cycles
- Move steps – specifies how often processing of the geometry has to be repeated in horizontal direction
- Move vertical – moves the grouped geometry along Y axis, this option can be combined with “Move horizontal” to define two-dimensional movements
- Move distance – the distance a movement has to be done in vertical direction between two processing cycles
- Move steps – specifies how often processing of the geometry has to be repeated in vertical direction
- Move radial – defines a movement operation onto a ring, this option can not be combined with one of the previous splitting direction modes
- Diameter – the total diameter of the ring where the geometry has to be marked at
- Move angle – the movement angle for every split, the size of the single split pieces result out of this angle and the diameter of the ring

- Move steps – specifies how often processing of the geometry has to be repeated on the ring
- Keystone correction – when marking the inner side of a ring the laser beam has to be slanted in order to hit only the lower inner side of the ring and not the upper outer side too. Resulting from that the length of the beam varies dependent on the X/Y position of the geometry which results in a distortion when no special optic is used.  
Using the keystone correction this distortion can be suppressed so that the marking result will look like expected. This correction is done by setting an as a percentage correction factor and by defining the side of the geometry this factor may apply to. The correction factor specifies how much the geometry shall grow or shrink according to its original size at the outer position where the full distortion has to be expected.
- Process order – this order applies to the planar movement modes and defines in which order the grouped geometry have to be repeated:  
Left to right – horizontal direction always starts from left  
Right to left – horizontal direction always starts from right  
Zig-zag from left - horizontal direction starts from left and performs zig-zag movements  
Zig-zag from right - horizontal direction starts from right and performs zig-zag movements  
Top to bottom – vertical movement starts from top  
Bottom to top – vertical movement starts from bottom  
 In case two-dimensional motion is defined both process order selections are combined with each other to describe the 2D motion.
- Motion speed: the movement speed in unit mm/sec or degrees/sec that is applied to the motion
- Axis mapping: at the lower end of the property panel a matrix of radio buttons exists where the axes can be mapped to the related movements, there it can be specified which of the axes A, B or C of the two possible outputs OUT2 and OUT3 have to provide the related planar movements in horizontal or vertical direction or the angular movement.
- Save as default – after application start-up new move-groups are created with some default parameters. When these parameters are changed, other move-groups then are created with these changed parameters, but after restarting BeamConstruct, the default parameters are used again. When this button is pressed, the current move-group parameters are saved as new default so that they are used whenever a new Active Move Group is created after BeamConstruct was started newly. The related data are stored within the users working directory in a file named “.beamuiMOVE”

After such a splitting group has been created and after the correct parameters have been set, the appearance of the related geometry does not change but within the entity list a different symbol for the group is shown in order to point to the fact a special kind of group is used here.

### 5.9.3 Edit Destructive

The functions of this category perform irreversible modifications to the input geometry. That means only the resulting geometry is used for further processing and all source elements that have been used to create this geometry get lost.

Resulting from that there is much less computing time and memory used to hold such geometry and editing on a per-vector-level is much less difficult because the data no longer can't be overwritten by a re-creation caused by some parameter changes. On the other hand the conversion to static geometry means that no more dynamic changes are possible like they can be performed when Data Input elements are used – as soon as these elements have been modified by such a destructive modification, all the (sub)elements it consisted of are lost. But afterwards it is possible to assign Additional Geometry and Post-processing elements to this static geometry again.

Following functions exist that perform such destructive modifications:

- Merge – the geometries of the selected elements are merged together, the result is one static element with all geometries included into it; this function can also be applied to one single Primary Geometry element and its sub-elements to reduce its memory usage to the absolute minimum so that only the resulting static data of it are stored
- Split – tries to split the geometry of an element into several pieces, here splitting is done at positions where a geometry ends (e.g. at the end of a logic element or the end of a polygon). There are

several levels of splitting so that it is possible to perform a splitting operation up to two times at the same data. As an example: when the splitting-function is applied to a geometry that represents the string "Text" for the first time, the result are four elements containing the geometries for the letters "T", "e", "x" and "t". In a second step the splitting function can be applied to the geometry of the letter "e" again. Now it is split based on the contained poly-lines, the result are two elements that contain the two polygons the letter "e" consisted of (one with the outline and one with the small part within the head of the "e").

- Optimize – this function checks geometry for inefficient structures and optimizes them. Here the visual representation of a geometry is not changed but the number of vectors and vertices is reduced if possible.
- Reduce Geometry – here the data a geometry consists of can be reduced. After selecting this menu item a dialogue is opened which provides different reduction methods and modes:
  - Smooth lines – this is not a real data reduction method but combined with the other options it can have stronger effects and better results; this expects a line-length in unit mm, when chosen all lines within a polygon that are smaller than this value are smoothed, means their values are combined in a way which tries to result in softer shapes with less edges (low pass filter); this function should be used with round shapes and can improve the effect of option "remove lines" when used in combination -
  - "Remove lines" - here a parameter in unit mm is expected which specifies the minimum length a single line within a polygon shall have, lines that are smaller than this are concatenated with their predecessor
  - "Remove points in angles" - this option expects a parameter in unit degrees and removes all points within a polygon which form an angle that is smaller than the specified value (means three points are inspected which form an angle and the middle point is removed when the angle is below the given threshold); the effect of this option becomes weaker when it is combined with function "smooth lines"
  - "Close polygon with endpoint" - this is not a real data reduction method but combined with the other options it can have stronger effects and better results; this function tries to create closed polygons out of a shape; when the end of a polygon line is found, it is checked if it is a closed polygon (means the end is located at the same position like its beginning) or not; in case it is not closed, the beginning of an other polygon line is tried to be found which is closer than the parameter entered for this function (in unit mm) – and when one could be found, the whole geometry is rearranged in order to get one continuous polygon; this function can improve the effect of option "remove lines" when used in combination
- Extrude – this function extrudes a flat, two-dimensional element into the third dimension. After clicking this menu item, a dialogue opens, where several options can be chosen which influence the extrusion result:
  - Extrusion axis – specifies along which axis the extrusion has to take place, this is done always parallel to one of the three axes X, Y or Z so that it is recommended to align and position the element properly before this function is called
  - Extrusion direction – specifies the extrusion direction (up or down, left or right, in our out)
  - Distance – in normal extrusion mode (3D mesh option not selected) this operation is performed by duplicating the element and by placing the duplicates close to the original. This value can be used to choose the distance between all these copies, the smaller is it, the more closed the resulting geometry is
  - Extrusion length – specifies the total extrusion depth, means how much the element shall grow into the selected direction
  - Create 3D mesh – when this option is set, the "Distance" parameter is no longer used. Here instead of a normal primary element a 3D Mesh is created which is part of a slice group (please refer to related sections below) and which can be used to modify existing 3D meshes
  - Close 3D mesh – this option can be used together with the previous one, when it is set, instead of an open "tube", the generated mesh is closed on top and bottom, forming a solid 3D model
- Boolean add / Boolean subtract / Boolean intersect – these functions perform boolean operations with selected primary elements. For such an operation at least two elements need to be selected. But in general several primary elements can be selected to perform one of these operations with. The "add" function concatenates the elements to form one single, closed polygon out of them. Overlapping parts will result in one closed shape afterwards. With the "subtract" function that element which is selected as second is used to cut from the shape of the first one. For this operation it is recommended to always use exactly two elements as the results may be confusing elsewhere.

The “intersect” function finds and returns these parts, where the selected polygons overlap each other.

- Project onto ground – this function is part of the possibility to have a 3D working area surface (as described in section “5.8.1.3 Drawing Area with 3D surface ”). It changes all selected geometries in Z-direction in a way, so that they are located on the ground surface
- Delete Element(s) – deletes the selected element(s)

## 5.10 Process 3D Data

There are two ways to process 3D data in BeamConstruct: one possibility is to use their 3D information including the Z (=depth) information to drive a 3D scanner and to mark them exactly as they appear. Alternatively a solid 3D model can be imported and cut into horizontal pieces (so called “slices”) which can be processed separately and optionally without the need to involve a 3D scanner system.

The first variant does not require any more than a scanner controller card that is configured for 3 axis operation. Resulting from such a configuration the user interface of BeamConstruct is switched into 3D mode too, now it is possible to edit Z-values within all property panels, to switch the drawing area to profile views or to a real 3D view and to draw and edit elements also in depth-direction.

The second variant can also be done with a 2D scanhead and a motion stage performing the Z-movement after every slice. This mode is named “2.5D”.

### 5.10.1 Slicing 3D Data

The second processing method (as described in previous section) does not necessarily require a 3D scanner, here a 2-axis scanhead such as the HALscan 10X20 can be used too, the third dimension can be handled by external hardware (like a drive that moves the position of the working piece in Z-direction via a lift table). As a precondition to use this operation mode a ready-to use 3D model has to be imported. Currently BeamConstruct supports the 3D Manufacturing Format 3MF format, the 3D Studio 3DS format, the Object File Format OFF, the RenderWare RWX format, the Surface Tessellation Language STL in binary and ASCII format, the WaveFront OBJ format and the PLY 3D printer format and others more. Beside of that the Common Layer Interface CLI format is supported too, since this is a file format that already contains sliced data it can be used for processing but is not a real 3D format where a slicing operation can be applied to.

More directly supported 3D file formats will follow in future. In case 3D data are available using a different file format than the ones described above, a converter application can be used. Here we recommend AccuTrans 3D which is available at <http://www.micromouse.ca/> and supports nearly every 3D file format that exists. AccuTrans 3D itself is available for an extremely low price.

After selecting a 3D file for import, a configuration dialogue opens where some initial slicing parameters can be configured.

- “Use as ground shape for working area” – when this option is selected, the 3D model is NOT imported for slicing. Instead of that it can be used as ground shape as described in section “5.8.1.3 Drawing Area with 3D surface ”
- “Slice into layers” – when this option is used, the 3D mesh will be imported for 3D processing and all the following values are valid

All the following slicing parameters can be changed later within the application:

- Import Scale Factor – a factor that can be used to scale the whole 3D object during loading
- Slice Distance – the distance between every slice
- Slice Offset – specifies an offset from top/bottom of the 3D model where slicing has to start; in case this value is set to 0.0 it may result in a first slice that contains nearly no relevant data (dependent on the used 3D model)
- Reduction – expects a value in unit degrees to perform a lossy data reduction on sliced data; whenever an angle between two vectors of a layer is smaller than the angle given here, they will be

concatenated to one common vector. For small angle-values and in most cases this results in a memory consumption which is dramatically lower and processing speed therefore much higher while there is no visible loss in quality of the result.

- Close lines – this parameter can be used to try to repair malformed or defect 3D models. When two not connected lines are located closer to each other than the distance specified here, the software tries to connect them. So this function tries to close holes in the imported 3D model. Please note: when the value is set to too big values, it may happen that the wrong open ends of slice lines are connected to each other – resulting in a damaged slice.
- Slice Direction – here the slice direction for the 3D model can be selected, “Top To Bottom” may be used e.g. for depth-engraving where operation starts on top of the model and moves into the material, “Bottom To Top” can be used e.g. for rapid prototyping applications where processing starts from bottom and builds up a model slice by slice
- Real 3D / Make use of Z-coordinate – when the scanner controller card is configured for 2D only (so no third axis is enabled), slicing of a 3D model is done in a way where the Z-axis always stays at 0. In this mode for Z-movements some additional actions are necessary such as adding a motion operation between each slice. In this mode, due to the lack of the third dimension, the “Real 3D” option is disabled and can’t be used.  
When the scanner controller operates in 3D mode (e.g. with a real 3D scanhead connected), the Z-coordinate can be calculated by the slicing function so that the model can grow in Z-direction controlled by the scanhead. To do that, this option has to be set. When operating in 3D mode but this “Real 3D” option is not enabled, the Z-coordinate is no longer used for building the 3D model and again stays at 0 and again requires a separate Z-shifting. This is useful e.g. in cases where the 3D scanhead is used for keeping the focus but not for doing real Z-movements.
- Split to tiles – the working area is split into (invisible) tiles that are used to rearrange the contours of the sliced 3D mesh during marking. Using these tiles it can be avoided to heat up the material at a specific position too much, it lets the position where contour is marked make additional interruptions and jumps to other place in order to continue marking at a distant position (means a distant tile). When this option is set, the following two parameters can be used to specify behaviour of this feature. For a more detailed description and usage examples please refer below.
- Tile Size – the size of a tile in working area, this value should be such a fraction of the working area size, that lets all tiles fit fully into whole working area
- Tile Skip Size – skip distance between tiles, this value decides how many tiles in horizontal direction should be skipped before marking continues in next tile; remaining tiles are processed on next cycle; this value should be much smaller than the total number of tiles that fit into the working area
- Clean between slices – when this option is set, a 3D Slice Cleaner element is generated on construction of the Sliced 3D Mesh Group. For more information about this type of element which can be used for cleaning purposes between slices, please refer to section “5.10.4 3D Slice Cleaner”

When this configuration dialogue is left by pressing the “OK” button the 3D model is loaded and sliced according to the specified parameters. Dependent on the size of the model, the number of slices that result out of this size, the slice distance and the complexity of the models geometry this operation may take a while. When the slicing operation is finished a “Sliced 3D Mesh” group element is added to the current project that contains the 3D model and the slices.

### 5.10.2 Sliced 3D Mesh Group

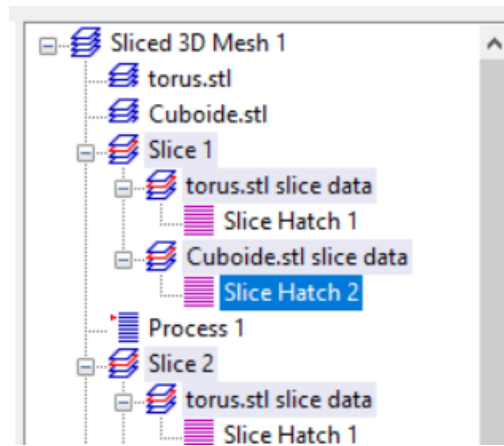
This is a very specific group that contains a 3D model that will not be marked and several slices which are generated out of this model. Such a group has to be combined with other elements carefully since it performs operations in Z-direction that have to fit to other elements or groups that are located before or after it within the current project. A Sliced 3D Mesh Group is shown as a group element with one 3D Mesh element for every imported 3D model, and one sub-element for every slice within the element tree on the windows right hand side.

When the group is selected a special property panel opens on the windows left hand side. There several parameters can be changed. Some of these parameter changes lead to a recalculation of the slices which means the currently existing slices, their sub-elements and possibly added motion-elements are removed from this group. So it is recommended first to optimise all the slicing parameters and then add more data and elements to such a group.

The Sliced 3D Mesh Group property panel itself is split into two major parts. Section "Model" contains all parameters that belong to the 3D mesh and the related data. Section "Support" is related to (automatic) creation of additional structures that are intended to help creating the model itself, especially at positions where parts of the model hang over and would not be stable during the phase of marking of single slices.

In section "Model" following parameters can be changed:

- Slice Distance – the distance between every slice; when this value is changed all slices and possibly existing additional elements are removed and the slices are generated again
- Slice Offset – specifies an offset from top/bottom of the 3D model where slicing has to start; in case this value is set to 0.0 it may result in a first slice that contains nearly no relevant data; when this value is changed all slices and possibly existing additional elements are removed and the slices are generated again
- Reduction – expects a value in unit degrees to perform a lossy data reduction on sliced data; whenever an angle between two vectors of a layer is smaller than the angle given here, they will be concatenated to one common vector. For small angle-values and in most cases this results in a memory consumption which is dramatically lower and processing speed therefore much higher while there is no visible loss in quality of the result.
- Close lines – this parameter can be used to try to repair malformed or defect 3D models. When two not connected lines are located closer to each other than the distance specified here, the software tries to connect them. So this function tries to close holes in the imported 3D model. Please note: when the value is set to too big values, it may happen that the wrong open ends of slice lines are connected to each other – resulting in a damaged slice.
- Slice Direction – here the slice direction for the 3D model can be selected, "Top To Bottom" may be used e.g. for depth-engraving where operation starts on top of the model and moves into the material, "Bottom To Top" can be used e.g. for rapid prototyping applications where processing starts from bottom and builds up a model slice by slice; when this value is changed all slices and possibly existing additional elements are removed and the slices are generated again
- Real 3D / Make use of Z-coordinate – when the scanner controller card is configured for 2D only (so no third axis is enabled), slicing of a 3D model is done in a way where the Z-axis always stays at 0. In this mode for Z-movements some additional actions are necessary such as adding a motion operation between each slice. In this mode, due to the lack of the third dimension, the "Real 3D" option is disabled and can't be used.  
When the scanner controller operates in 3D mode (e.g. with a real 3D scanhead connected), the Z-coordinate can be calculated by the slicing function so that the model can grow in Z-direction controlled by the scanhead. To do that, this option has to be set. When operating in 3D mode but this "Real 3D" option is not enabled, the Z-coordinate is no longer used for building the 3D model and again stays at 0 and again requires a separate Z-shifting. This is useful e.g. in cases where the 3D scanhead is used for keeping the focus but not for doing real Z-movements.
- Use pen from 3D mesh – when this option is set, all the other parameters for pen configuration across the different slices (as described below) are disabled. Instead of that the pen that is assigned to the 3D mesh is used for both, all the slices contours and their hatches. This function is useful when different (small) models have been imported and each of them has to be marked with different pen settings and also different hatch patterns. The typical procedure to use this feature is as follows:
  - import a first 3D model via menu item "Import"
  - repeatedly right click the resulting 3D slice group in element tree and select "Import 3D model"
  - go to tab-pane "Element" and select a pen for each of these 3D meshes
  - select the option to use the pen from the mesh and start slicing
  - in element tree open the slice element by clicking the small "+"-sign
  - each of these slice elements contains a slice data element for each imported 3D mesh
  - select each slice data element and add a hatch to each
 Now the element tree will look similar to this:



Now each 3D mesh corresponds to a separate slice data element where its own hatch element is added (and even more can be added).

- Slice Pens – this table can be used to define patterns for the pens that have to be assigned to every slice. There separate pens can be specified for contour and hatch of a slice. First of all the check box in front of a row of the list has to be selected to enable this entry. Then the pens can be selected via the combo box. Here the list rows have different meanings:
  - “First Slice” is used for the very first slice only (that can be on top or bottom of the model dependent on the slice direction)
  - “Last Slice” is used for the very last slice only (that can be on top or bottom of the model dependent on the slice direction)
  - 1..5 (all other rows) within that list define an order of pens that is repeated until the last slice is reached. Here patterns of up to 5 different pen combinations can be defined. Each pen definition (=row of settings) comes with an additional input field “Repeats” which defines how often the current contour/hatch pen combination has to be repeated before the next row is processed. Using this function it is possible to have more than one slice with the same pen definition before switching to the next variant.
  - Din – when this option is set to a value different than “Off”, an override-function is enabled. Here “Din” defines a digital input on the used scanner controller card. As soon as this input is set to “HIGH” during marking, none of the pen-definitions “First”, “Last” or “1..5” are used for marking, but the pens for contour and hatching that are defined within this row. When the input is “LOW” during marking or when the digital input function is set to “Off”, the regular pen definitions are used. When a scanner controller card comes with different digital input ports, the digital input bit is read from the first port according to this order: Digital 1 (8 Bit), Digital 2 (8 Bit), Digital 1 (16 Bit), Digital 2 (16 Bit), Digital (32 Bit)

Since the pen-definitions for sliced geometry have to be done here, the related option within the “Element” tab-pane is disabled, pens can be assigned only via this pen mapping table.

- Show All Slices – in case this check box is selected all slices that belong to a 3D model are shown within the drawing area of BeamConstruct. This function has to be handled with care since a large amount of slices can slow down the application dramatically, here it is recommended to use “Display reduced” parameter to show only every nth number of slice
- Hide hatch from view – A hatched model can – dependent on the number of hatch lines calculated for a slice – slow down graphic visualisation quite dramatically. For such cases it is possible to enable this option which then disables drawing of the hatch patterns of a 3D slice group in order to have a faster and more smooth graphical representation of the current project
- Slice during marking – normally complete slicing of a model is done in advance and for the complete model. Depending on the complexity of a model such an operation can consume quite a lot of time while no other operations are possible. When this option is checked, only the very first slice is precalculated and all other ones are done during marking: while a slice is marked and the related inter-slice operations are done, the next slice is calculated and the related hatch-patterns are applied

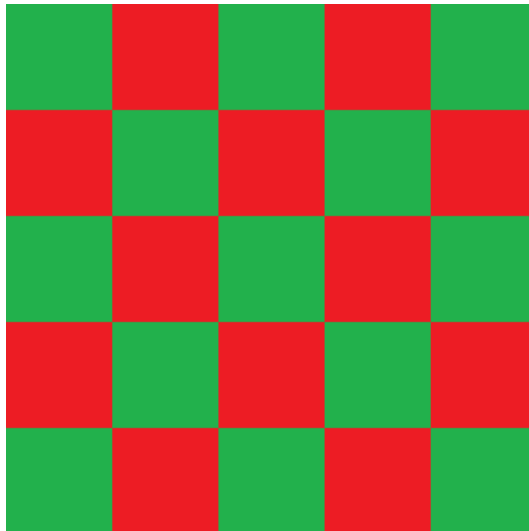


in parallel. Thus a lot of handling time can be saved since both operations – marking of a slice and completely preparing the next slice – are done fully parallel.

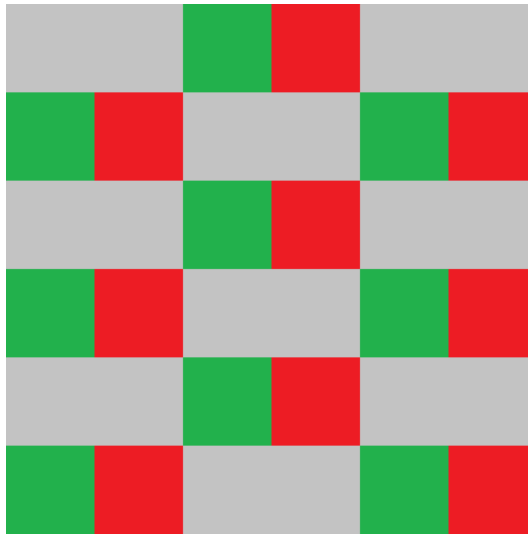
This function can be used only with models where no support-structures are generated, means activating this option automatically disables generation of support-structures.

After setting this option, slicing and adding additional elements to the generated slices has to be done exactly in the same way as before. But with this option set, only the very first slice is shown. All others are generated during marking and therefore are not available in advance.

- Display Reduced – reduces the number of slices that are shown in case option “Show All Slices” is selected, the bigger this number becomes the less slices are shown within the drawing area
- Split To Tiles – when this option is set, the working area is split into (invisible) tiles that are used to rearrange the contours of the sliced 3D mesh. Using these tiles it can be avoided to heat up the material at a specific position too much, it lets the position where contour is marked make additional jumps in order to continue marking at a distant position (means a distant tile). When this option is set, the following two parameters can be used to specify exact behaviour of this function. After enabling this option the 3D mesh is re-sliced but there will be no visible representation of the changes in working area, it can be noticed during marking only. This function influences only the contour of the slices, hatches are not interrupted and broken into tiles in order to avoid damaged marking results. To reduce heating of material during hatch processing, please refer to description of the hatcher above; there exists a similar functionality to skip or to tile hatch lines. Together with the following parameters this function can be used to e.g. mark every second tile:



Assumed the working area has a size of 100 mm x 100 mm, for this behaviour "Tile Size" needs to be set to 20 mm and "Tile Skip Size" has to be set to 2. With these settings first all these contours are marked that have vectors within red tiles, finally all the remaining contours that have a vector in green tiles are marked. Tiles itself are processed from upper left to lower right corner row by row. It is also possible to have even bigger distances between tiles so that they do not touch at their edges:



Parameters for a 90 mm x 90 mm working area are 15 mm for "Tile Size" and 4 for "Tile Skip Size". This results in a behaviour where marking starts with these contours that own vectors in red tiles, next the contours in green tiles are marked. This is continued until the whole working area was processed (in this example by processing the remaining two tile sets in grey areas with two additional steps).

- Tile Size – the size of a tile in working area, this value should be such a fraction of the working area size that lets the tiles fit fully into whole working area
- Tile Skip Size – skip distance between tiles, this value decides how many tiles in horizontal direction should be skipped before marking continues in next tile; remaining tiles are processed on next cycle; this value should be much smaller than the total number of tiles that fit into the working area

Section "Support" offers the following options and parameters which are available only when a model is created from bottom to top (so this option is not available when doing deep-engraving with a Sliced 3D Mesh Group):

- Enable Support Structures – this checkbox has to be enabled in order to let the software automatically create support structures, when it is turned off, the model will be processed as is and with no additional geometries. When enabled, additional geometries are calculated and added to the model automatically at positions where an overhang could cause problems. These additional geometries are only contours and need to be extended by hatch-elements in order to get them filled and stable. Here for both, contour and hatch suitable pens and related pen-parameters have to be chosen in order to not to melt together the support structures and the model's structures.
- Distance to Model – here a distance can be specified that always has to be used at positions where the contour of model and support structure are neighbouring within a slice; using this distance it can be ensured the support structure can be removed easily after processing. When this value is too large, the effect of supporting the model may fizzle out.
- Enlarge Structure – the value given here can be used to make the support structures larger than necessary in order to overlap the model completely; this value has no influence on the distance to the model, it expands the support structure to its outer borders.
- Style of support structures – the type of the support structure to be generated can be selected with this combobox, here following options are available:  
Enclose model completely – when this option is set, the support structure is created always, will cover the whole model from top to bottom and will exist anywhere independent from the fact if there would be some gravity effects making it necessary to have support structures  
All-over structures – the support structure is created from bottom to top until it is covered and replaced by parts of the model itself, working like a shadow that would be cast by the model itself. So in this mode support structures are generated at these parts of the model only, where negative effects caused by gravity are to be expected. Here no support structures are generated when there are (stable) structures of the model itself available. The generated support-structures are all-over, solid areas.

- Down to bottom of editing area – when this option is set, support structures are not only generated until bottom end of the 3D model but down to the lower end of the whole editing area (which in this case should be located below of the lower end of the 3D model)
- Support Pens – this table can be used to define patterns for the pens that have to be assigned to every slice's support structure. There separate pens can be specified for contour and hatch. First of all the check box in front of a row of the list has to be selected to enable this entry. Then the pens to be used for contour and hatch of a support structure can be selected via the combo box. All rows within that list define an order of pens that is repeated until the last slice is reached. Here patterns of up to 5 different pen combinations can be defined. Since the pen-definitions for sliced geometry have to be done here, the related option within the "Element" tab-pane is disabled, pens can be assigned only via this pen mapping table. Alternatively the pen-definitions for hatches can be disabled. To do that, the checkbox above the "Hatch"-column has to be unset. Now an own pen can be assigned to every hatch via the common way (pen-assignment in "Element" tab-pane). The difference between both methods: while pen assignment via the table in "Support"-section assigns the same pen to all hatches of a slice (which additionally may alter over slices), the pen-assignment via "Element" tab-pane can be used to assign different pens to each individual hatch that stay the same over slices.

Beside of that there is a slider available on the property panels left hand side: there the currently shown and used slice can be selected and it can be used to scroll through the sliced model. The same is true for the input field below of that slider: there a number of the current slice can be entered or chosen.

Such a Sliced 3D Mesh Group is shown as a group element with one sub-element for every slice within the element tree on the windows right hand side. Comparing to other groups it behaves a bit different. So it is possible to put new elements (like a motion control element or I/O elements) into that group directly. To do that, either the group node itself or one of its slices has to be selected and the Primary Element to be added has to be chosen within the tool bar. Now a dialogue is opened where the user can select if that element has to be added as a single one only, below of every slice or for a range of slices. This way it is possible to set motion controlling elements for every slice easily. More than this: all elements that are added via such a range will belong together afterwards: whenever a parameter of one of the elements is changed, the parameters of all others that belong to the same range (and therefore have the same number in its name) are modified too.

The same is true for Additional Elements: when there is a hatch added to a single slice the user can choose to apply it to a range of slices or to all slices. When the parameter of one of these hatches is changed the change is applied to all the other hatches that have been created together with this first one.

Using this way the same operations and parameters can be applied to all slices easily.

Between every slice-subelement of the group there is also a "Process"-element created automatically. This element does expect only one parameter: the path to an existing project file. This project file is executed at the position of every "Process" element and can contain processing-operations that are necessary between every slice (such as motion operations to move a Z-table, setting of digital outputs and waiting for digital inputs a to synchronise with external devices). Using such an external project it is not necessary to add these control-steps – which normally are always the same for a machine – manually for every newly sliced element.

### 5.10.3 3D Mesh

This element holds the imported 3D models which are used for processing within a Sliced 3D Mesh Group. When selected, it offers the following options in its related tab-pane:

- Show 3D Mesh – this option does not have an influence on the marking result but changes the behaviour. By default the 2D views do not show the mesh but only its outline. When this option is set, also in 2D views the full 3D model is drawn. As this can slow down displaying speed on some computers, this option is off by default and should be enabled only in case of a proper graphics hardware.
- Support structure model – an imported 3D model is set as payload object by default. When this option is set, the related 3D model is handled as precalculated support structure, means it uses the related pen parameters for such structures and ends up in the support structure subelements after slicing. Using this function it is possible to not only import precalculated 3D models but also the support structures which belong to them.

#### 5.10.4 3D Slice Cleaner

This kind of element typically is used in case of deep engraving applications but is flexible enough to be used in other areas too. It is not related to any geometry directly. When set within a Sliced 3D Mesh Group, during processing of the slices it is executed from time to time and can be used to clean up the current working area. The geometry of this element consists of the outline of the concatenated previous slices since the last slice cleaner was executed. Combined with Hatch elements (that can be added freely to it), it can be used to clean exactly that area that was processed during the last slices. The 3D Slice Cleaner and its hatch sub-elements all can make use of own pen settings.

Following parameters and settings are available in the properties tab-pane of this element:

- Clean every x. slice – defines the number of slices after that a cleaning cycle has to be performed. Here “x” specifies the number of slices which are processed without interruption and where the geometries are concatenated. Then after the xth slice has been finished, that geometry which is the sum of the previous slices is used to perform a clean-cycle – using the pen settings and hatch which is assigned to the 3D Slice Cleaner element.
- Derive cleaning shape from – here a mode can be chosen how the cleaning shape has to be created. Following options are possible:
  - Concatenated slices – in this mode the shapes of all slices are concatenated so that the resulting cleaning shape is the sum (aka “boolean add”) of all slices which have been processed until current position; the resulting, current cleaning shape can be extended in size via the “Enlarge by” parameter (please refer below).
  - First slice – when this mode is selected, only the very first slice is used to generate a cleaning shape from for the complete 3D model. When the first slice has no or a too small geometry, the next slice which has a valid geometry is used. The resulting, current cleaning shape can be extended in size via the “Enlarge by” parameter (please refer below).
  - Generic rectangle – instead of a complex shape which is calculated out of the slices of the related 3D model, in this mode a simple rectangle is used as cleaning shape. The rectangle itself is the 3D Slice Cleaner element, means it can be positioned and dragged directly within the drawing view or its size and position can be entered in the tab-pane “Geometry”. So in this mode editing of the rectangle to be used is possible in the same way like dragging and scaling can be done like for any normal element. Please note: rotation of the shape is not supported yet.
  - Generic circle – instead of a complex shape which is calculated out of the slices of the related 3D model, in this mode a simple circle is used as cleaning shape. The circle itself is the 3D Slice Cleaner element, means it can be positioned and dragged directly within the drawing view or its size and position can be entered in the tab-pane “Geometry”. So in this mode editing of the circle to be used is possible in the same way like dragging and scaling can be done like for any normal element.
- Enlarge by – this parameter is related to the concatenated geometry of the cleaner-element and is valid for modes “First slice” and “Concatenated slices”. The value given here in mm specifies the amount this geometry has to be enlarged in order to cover an area which overlaps the one which was used by the slices effectively.

#### 5.10.5 Auto-Arrangement of 3D Meshes in Sliced 3D Mesh Groups

A Sliced 3D Mesh Group can hold several 3D Meshes. They appear as sub-elements of type “3D Mesh” in element tree. These 3D Meshes can be positioned with the standard functions in tab-pane “Geometry” in order to place them at suitable positions in working area. Alternatively it is possible to let BeamConstruct arrange these meshes automatically. To do this, the Sliced 3D Mesh Group has to be selected and right-clicked in element tree, then menu item “Auto-Arrange 3D Meshes” has to be selected. Now a dialogue opens which provides options to arrange the 3D Meshes in X- and Y-direction and to align them at a specific Z-height. This is done via the configuration possibilities within the dialogue:

- Arrange in X and Y – when enabled, the 3D Meshes are arranged on working area using the following parameters
- Distance to borders – specifies the minimum distance the 3D Meshes need to have to the border in both, X and Y direction of the working area

- Distance between meshes – specifies the minimum distance the 3D Meshes need to have between each other
- Arrange in Z – when enabled, the 3D Meshes are aligned at a specific height using the following options and parameters
- Align geometries at top or bottom – here it can be selected if all the meshes need to be aligned at their upper or lower position
- Arrange in Z – this is an option which can be enabled when 3D Meshes have to be top or bottom aligned at a specific, absolute height; when it is set, the height has to be specified in unit mm; in case this option is not set but arrangement in Z is enabled, all meshes are aligned at the highest/lowest top/bottom position of that 3D Mesh, that is located at the highest/lowest position (according to its outlines top/bottom border)

## 5.11 Vision

BeamConstruct provides an integrated Vision system for automatic adjustment and correction of marking data. Using this vision system special markers – so called fiducials – can be taught. Later the Vision system is able to recognise such a fiducial on a working piece and adjust position and rotation of the current marking data according to the fiducials measured position. Vision functions can be found in menu “Edit” submenu “Vision” and in mark dialogue.

To use the vision system it is mandatory to connect a camera and to configure a Video Controller in hardware settings. This enables the Vision menu entries and functionalities.

### 5.11.1 Vision Functions

Menu Vision → Start Video

When image capture is configured to be started manually this menu item can be used to start (menu item checked) and stop capturing (menu item unchecked)

Menu Vision → Freeze Video:

As soon as a Video Controller is configured, a live image is shown in background of the working area. This image changes permanently and can be frozen using this menu. This is a toggle menu, selecting it turns freeze mode on, selecting it a second time turns this mode off and enables the live-image again.

Menu Vision → False Colours:

In some cases captured images have a very low contrast in desired parts of the image. In such cases the shown image and the required details can be improved and made more visible by false colours. The sub-menu below of this menu item offers several false colour modes that can be used for this

Menu Vision → Calibrate Camera:

This functionality can be used to calibrate a camera and to adjust and correct distortions caused by the lens of the camera. When selecting this menu item, a calibration dialogue opens. For camera calibration following steps have to be performed:

1. A black/white chequerboard has to be placed in cameras view. Such a board can be found e.g. at <http://http://halaser.systems/checkerboard.png>. Here the board must be visible in cameras live image fully, none of the squares have to be cut.
2. The camera calibration dialogue has to be opened and the boards parameters have to be entered: here the number of crossing points between squares (the chessboards inner corners, not the number of squares itself!) have to be entered for horizontal and vertical direction.
3. The button “Try calibration” has to be pressed. It starts a calibration run where the chequerboards inner corners are searched. If this is successful, all these crossing points are marked by blue circles and the “OK”-button of the dialogue is enabled.
4. The dialogue has to be left with “OK” in order to take over the calibration data. Now they can be saved by selecting menu “Project” → “Save as default configuration”.

#### Menu Vision → Crop Camera:

Here it is possible to clip the camera image, means to cut the borders of the captured image to limit the view to these parts, that are really needed. After selecting this menu item a dialogue opens where the horizontal and vertical crop values can be set – the result is shown via a preview, that contains blue lines marking the new borders of the image. Cropping of an image takes place after a (possible existing) calibration was applied.

#### Menu Vision → Drop Calibration:

The current calibration data are dropped, afterwards the shown camera image is uncalibrated

#### Menu Vision → Teach Fiducial

Here one or more fiducials can be taught that have to be recognised later. This can be done in fiducial teaching dialogue which opens after selecting this menu item.

This dialogue shows the current – probably calibrated – image. There an area can be marked by drawing a rectangle directly into that image. After releasing the left mouse button a fiducial detection is performed within that area. The result of this operation is a list of keypoints which are marked by circles within the image and listed on the dialogues left hand side. Now within that list all these keypoints have to be removed (by pressing the “Delete” key or by using the related button) that do not belong to a fiducial (e.g. because marking of the area within the image was inaccurate or because they are located between desired positions when more than one fiducial is marked). During all that time a fiducial detection is performed with the same image to get a rough recognition quality value. This value is shown on lower left side of the dialogue. It should be as close as possible to 100%. When too much keypoints are removed, this values decreases – and the expected detection accuracy decreases as well.

This process of selecting image-areas that are of interest for the detection can be repeated several times, each time an area is selected, the resulting data are added to the list of keypoints. This way it is possible to mark several, non-rectangular areas within an image.

As soon as this dialogue is left by pressing the “OK”-button, the fiducials are taught and can be used. These fiducials can be saved in a separate file and can be loaded as needed with the following menu items.

#### Menu Vision → Load Fiducial

This menu can be used to load an already existing, saved fiducial definition. Currently used fiducial(s) is/are replaced by this function.

#### Menu Vision → Save Fiducial

This menu can be used to save the current fiducial definitions for later usage.

#### Menu Vision → Drop Fiducial

The current fiducial definitions are dropped, no more fiducial recognition is possible until new fiducial(s) is/are taught or loaded.

As soon as there is a fiducial definition active and valid, it is used for marking operations. When the marking dialogue is opened, a fiducial recognition is performed automatically. When the fiducial(s) can be detected in current live image, the orientation and position of the actual marking project is changed according to the fiducial(s) orientation and position. This procedure is repeated after every marking process automatically. Additionally it can be triggered again in mark dialogue by pressing button “Auto”, it starts a new cycle of recognition and correction.

### 5.11.2 Camera Calibration Cookbook

Sometimes the image shown by a camera may be distorted for different reasons. This may be caused e.g. by the optics of the camera or when it is not mounted 100% parallel to the working area. This may result in a somehow stretched and distorted image. Using camera calibration function this can be adjusted. To get an accurate and exact camera calibration is not an easy thing and may require a lot of trial and also some experience with such systems.

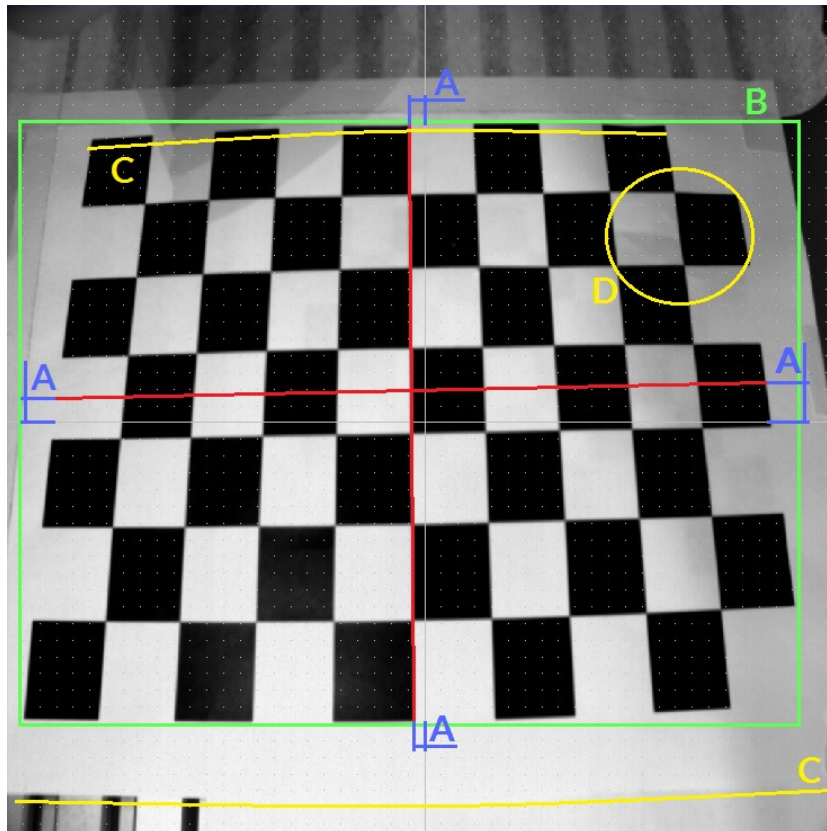
To make this easier, here some steps are described which should help to get an accurate calibration:

1. Print out the calibration template from <http://halaser.eu/checkerboard.png> or – when it does not fit to your visible working area size and aspect ratio – create an own one consisting of black and white squares. For printing an as thick as possible paper should be used so that it can be placed on the

working area without being bent, warped or folded. If necessary stick the checker board on a flat sheet of metal or an other stable material.

2. Use a camera with a high resolution. The checker board from the calibration template should be visible very clear.
3. Adjust the camera parameters in BeamConstructs hardware settings. First of all a monochrome image should be used. The vision system does not make use of colours and would convert the image to monochrome representation in every case. So enable greyscale mode in camera plug-in and check the points 1) and 2) again. Next it is recommended to use the native hardware resolution of the camera and to avoid to scale the image to a different size. Also keep the native aspect ratio of the camera.
4. Place the checker board template in cameras view, it should be located exactly centred, parallel to the borders of the camera, it should fit to the whole visible area completely. Beside of that it should not be bent.
5. Go into camera calibration dialogue, enter the number of visible checker board cross points and try a calibration.
6. Now crossing points all should be marked with a circle. When some crossing points of the checker board do not contain such a circle or when at least one of them does not fit to the real position of the crossing point exactly, the calibration recognition has failed. In this case you should adjust lighting of the calibration template as well as brightness and contrast settings until all crossing points can be detected exactly.
7. Now you can leave the camera calibration dialogue by pressing "OK" and you should be able to see a corrected camera image.

Following image shows some typical mistakes that would lead to a camera correction that damages the resulting image instead of correcting it:



A (blue) – the checker board is not aligned properly, it is rotated and shifted. Here the red lines mark where the crosshair from BeamConstructs working area should be located at really. The calibration template in this image is shifted to the left, to top and rotated to left. Since camera calibration system does not know this is the template only, it will assume this is caused by the camera and will try to compensate it.

B (green) – marks where the visible area of the camera really should end and how the calibration template should cover the cameras visible field. If necessary, an own checker board has to be created that has the required aspect ratio and that contains the required amount of squares. The more squares that are visible, the more exact the calibration becomes.

C (yellow) – here the calibration template is bent, it is not placed flat on and parallel to the working area. Here the camera calibration function will also try to compensate this bent part of the image – which will result in an overcompensated image since the real working area is not bent in the same way.

D (yellow) – the calibration template is folded. The calibration function will try to compensate this damage too (which is in template only but not a camera-caused distortion).

### 5.11.3 Fiducial Calibration Cookbook

To get a vision system with a fiducial recognition that is working well and that produces exact results is not an easy thing and requires a lot of adjustment and experience. As a starting point and to get some basic information to make own experiences with, following some hints are given to get better results. The steps listed below should be performed as accurate as possible:

1. Use a camera with a high resolution. The fiducial(s) should be visible very clear and quite large so that all relevant details of it/them can be seen in cameras image.
2. Adjust the lighting. As first step a working piece should be placed in cameras view. Now adjust the light so that the fiducial(s) on that working piece can be seen clear and has/have a good contrast comparing to its underground and comparing to other markers on this working piece. Think about using coloured light which could result in better contrast between fiducial and background.
3. Adjust the camera parameters in BeamConstructs hardware settings. First of all a monochrome image should be used. The vision system does not make use of colours and would convert the image to monochrome representation in every case. So enable grayscale mode in camera plug-in and check the points 1) and 2) again. Next it is recommended to use the native hardware resolution of the camera and to avoid to scale the image to a different size.
4. Calibrate the camera as described above. Please note: in case your camera has a really good lens, does not distort the image and is mounted parallel and on top above the working area, this step should be dropped. The camera calibration always results in a loss of information and quality and therefore should be avoided when possible. Nevertheless: a lens distortion is much more worse than this loss of information, so this should be dropped only in case of a really good camera!
5. Teach the fiducials as described above. It is not possible to make any forecast which of the fiducial recognition keypoints has which influence on the recognition result. Thus several iterations are necessary here: select the fiducial(s) in teach dialogue, remove the unused keypoints, accept the result and try recognition with a changed position of the working piece. And depending on the results: start from the beginning with fiducial selection and keypoint removal.

## 5.12 Processing Operation

To manipulate a loaded project related to the processing operation several functions can be found in menu "Process" of the main window:

- Simulate – a full process cycle is simulated, means the complete project is updated, all serial numbers, date and time objects are triggered, the process time of geometries is calculated, delays are measured and movements are (partially) simulated. The resulting total process time is displayed to the user. This time is nearly the same time such a process would need within the target environment. It differs only in case motion elements are used that perform absolute movements, here the real processing time can't be calculated because the starting position of a driven axis is not known.
- Mark: this option requires at least a scanner card to be configured within the global project settings and gives the possibility to start a real marking process instead of simulating it. In case the current project uses motion elements the related motion controllers should be configured too within the



project settings.

After selecting this menu item a marking dialogue opens, for more information about its functionalities please refer below.

- Write Stand Alone Data – this function applies only to these scanner controllers that support the related feature; it gives the possibility to write scanner data to a disk for later usage in stand-alone mode. When this function is used in a multihead-setup (please refer to “5.16 Multihead Operations” for additional information) where all controllers support this function, multiple files are generated. For all additional files, the number of the related head is appended to the file name.
- Send Stand Alone Data – this function applies only to these scanner controllers that support the related feature; it gives the possibility to send unidentifiable scanner data to the controller without starting a mark operation immediately but for later usage in stand-alone mode
- Send Named Stand Alone Data – this function applies only to these scanner controllers that support the related feature; it gives the possibility to send named scanner data to the controller without starting a mark operation immediately but for later usage in stand-alone mode. Comparing to the preceding option this one opens a dialogue where following data can be entered:
  - the name the data are stored with at the scanner controller card and which can be used later to select this specific project in stand-alone mode; the rules this name has to be created with depend on the scanner card
  - an option “Send full data to all cards” which is enabled in multihead mode; when it is set, it can be used to copy the full vector data to all configured controller cards, elsewhere only these data are to every card, which really belong to it according to the multihead marking rules
  - an option “Reset controller after download” which reboots the controller after downloading has been finished; this option can be useful when data have been updated on the controller which are somehow pre-buffered and need to be reloaded on boot. This option requires a hardware and plug-in which supports rebooting of the connected device. If no reset-functionality is available for a scanner controller, this function is ignored.
- Show State – this option shows different scanner controller state information; which information are shown depends on the used scanner card and its capabilities but it may consist of
  - digital output data (reflects the data that have been sent to the output but not necessarily the data that really are set as long as the card does not support read back of these outputs)
  - digital input data
  - analogue input data
  - plug-in information
  - scanhead information
- Increment – all serial number elements of the current project are incremented, the date and time elements are updated
- Decrement – all serial number elements of the current project are decremented, the date and time elements are updated
- Reset – all serial number elements of the current project are reset to its starting value, the date and time elements are updated

## 5.13 Marking

The mark dialogue can be opened via menu “Process” menu item “Mark”. When this dialogue is opened for the first time or in case the hardware settings have been changed after it was used the last time, as first operation the scanner controller cards and the – optionally configured – motion devices are opened. Motion axes that are configured to reference on start-up are moved to its reference/home position. All these operations may take some time dependent on the connected hardware.

In case this could be completed successfully, the dialogues mark-buttons can be used afterwards. It provides a button for starting the pilot laser (requires fully configured pilot laser outputs in project settings) on the left side of the dialogue, a mark-button in the middle that starts the main laser and a stop-button on the right side that stops the main or pilot laser. Below of them some options can be found that influence the marking operation:

- Mark repeatedly – starts an infinite loop, as soon as the mark-button is pressed the current geometry is marked again and again until the stop-button is pressed
- Fast repeat – this check box becomes active only if a) “Mark repeatedly” is set and b) all used scanner controller cards within the current configuration support this function. When it is set, it gives the possibility to have a very fast loop without any delay between end of previous and begin of current mark cycle. This function buffers several mark-cycles on the scanner controller card in advance and is intended to be used with short projects which have to be marked very fast after each other (e.g. when marking of these projects is started by external triggers following in short time distances after each other).
- Mark selected elements – when this option is set only these parts of a project are marked that are selected within the Entity-Tree and highlighted within the drawing area
- Wait for external trigger – using this option marking is started only under two conditions: first the start-button is pressed, next an external trigger signal is applied. Together with the “mark repeatedly” function this can be used to start marking via an external hardware signal and no longer by pressing the start-button repeatedly.
- Suppress MOTF – in case of marking on-the-fly applications a project may be designed in a way where the external trigger signal is mandatory, elsewhere the project will not work. But exactly this behaviour which is wanted during normal operation, can be a problem when a project is adjusted as it will not work and not show what is marked. So checking this option, marking on-the-fly functionalities are inhibited so that a project is marked also without external encoder pulses being available. This mode is for testing a project only, and of course it will damage the layout when it is designed in a way where it does not fit to the working area completely. Nevertheless with this option temporarily unwanted MOTF-functions can be turned on and of fast and easy

Additional marking-related functionalities are arranged in separate tab-panes that are grouped according to the functionality they contain. tab-pane “Pilot Laser” contains functions for accessing the preview/pilot-laser that can be started by the upper left mark button:

- Mode – this combo box gives the possibility to choose the pilot laser marking mode, “Off” disables usage of the pilot laser “Manual” expects that the user starts the pilot laser manually by pressing the big pilot-laser-button on the windows upper left hand side and “Auto” re-starts the pilot laser after every marking process automatically. Also in mode “Auto” the pilot laser has to be started manually for the first time because of security reasons.
- The radio buttons below this combo box can be chosen to select what the pilot laser shall show in its preview, “Use total projects outline” draws a simple rectangle that includes the whole project, “Use individual elements outline” draws a simple rectangle for every element within the current project (except dots, lines and triangles which are shown fully) and “Use real element shapes” draws the full shape of all elements except hatches.  
The pen – and therefore the related speed and delay values – that is used for the pilot laser can be defined within the pen settings dialogue (please refer above).

Tab panel “Manual” offers direct access to the position of scanhead. Here mirrors can be moved to a specific position:

- In upper combobox the scanhead can be chosen this operation has to be performed on (valid only for multihead configurations, elsewhere “Scanhead 1” has to be used)
- four arrows give the possibility to move the scanner position in positive and negative X- and Y-direction; the input field between these arrows specifies the jump length every button press has to cause; pressing these buttons lets the scanner jump immediately
- In fields “X” and “Y” coordinates can be selected the scanner has to be moved to, by pressing button “Jump” motion to these positions is started
- The button with the camera symbol in upper right corner of this tab-pane gives the possibility to toggle a live-background image (in case an image capture device is available and it is configured to be shown within the drawing area). On weak computer systems this function can be useful to turn this live view on only as long as it is needed to not to slow down or disturb the main marking process.
- Below of the camera-button there is a small pilot-laser-button which gives the possibility to turn on and off the pilot laser (only available in case a pilot output is configured in hardware settings).

Together with the coordinate buttons it can be used e.g. for referencing purposes – when the pilot laser permanently points to a fixed, defined position which also can be changed as needed.

In tab panel Geometry the current projects vector data can be manipulated:

- Arrows left/right/up/down – using these arrows the whole geometry of the project can be shifted into the given direction (in top view for 3D mode); the shift step size in unit mm is taken out of the input field below of these arrows
- Arrows rotation left/right – using these arrows the whole geometry of the project can be rotated into the given direction (in top view for 3D mode); the rotation step size in unit degrees is taken out of the input field below of these arrows
- Button Auto – this button can be used only when a live image is captured and when fiducials are taught using the integrated vision system, it tries to detect the taught fiducials in current image and aligns and rotates the current geometry automatically according to the position and orientation of these fiducials
- Button Reset – when this button is pressed all changes in position and rotation of the current projects geometry are taken back, it is reset to its default values

In tab panel Motion it is possible to drive configured motion axes manually and independent from any motion elements in marking project. Please note: when there are some marking elements used directly within the project, it has to be taken into account that the user may have changed the position of the axes via these manual functions. Thus at least the first motion element should perform a movement to an absolute position or should reference its position. Within the mark dialogue following motion functions are available:

- OUT2 / OUT3 – specifies which of the two possible motion controllers has to be used
- Speed – specifies the speed a manual motion operation can be performed with;  
PLEASE NOTE: when the speed value given here is bigger than the maximum speed of the related axis, this value is clipped internally, means the axis moves with it's maximum speed which may be slower than the nominal speed entered here
- Camera – this button can be used to toggle the background image on and of in case manual image capture enabling is configured within project settings; this button has the same functionality like the related vision menu item
- movement value – the value that can be entered in middle between the arrows specifies the relative movement distance or the absolute motion position to move to (depending on the arrow button that was pressed)
- Double arrows – these buttons cause a movement as long as the button is pressed
- Single arrows – these buttons cause a single movement via the distance or angle specified with the number input field in the middle, here the time the button is pressed does not influence the motion
- Arrow to dot – when this button is pressed an absolute movement is performed, the number input field in the middle between the buttons specifies the target position to move to, the time the button is pressed does not influence the motion
- Ref – start referencing of the related axis

For planar movement axes the images of the buttons can be changed within the motion axis configuration panel of the general project settings. Depending on the real movement direction (left/right, up/down, in/out) a different symbol can be shown here in order to make it easier for the operator to decide which direction a motion will use. The same is true for the axis names, they can be configured via the axis alias value in general project settings.

The third tab panel Parts is related to parts and part counting, some of its buttons and functionalities will be enabled only when part counting is enabled for the current project (can be done in general project settings dialogue). It offers the following functions:

- When part counting mode is enabled the large counter shows the current number of parts already marked. When the limit specified in project settings or in input field "To produce" is reached, no more marking operations are possible until the user presses button "Reset Counter"
- The "Serial numbers" buttons can be used to change the state of all input elements of the current project for the next part to be marked, here "Increment" increments all of them by one, "Decrement"

decrements them by one step and "Reset" sets them back to their start value. This is the same functionality like it is accessible from within menu "Process".

The tab panel "Progress" shows some information regarding the current marking progress, its functionality is limited to some specific applications. First of all it is feed with timing and operational data only when sliced 3D marking data are processed. Next it will not show the real marking state of the connected controller but only the state of data sent to the scanner card. So when a scanner controller card with a large buffer is used, it may hold some none-processed data while for this progress bar they are already finished. So this progress information is mainly suitable for long-running processes (where relative deviation between mark-dialogue-estimated and real marking time is quite small) but not for getting a process forecast for fast marking operations. For last case it is still recommended to use the function "Simulate" in menu "Process". Beside of that it offers two buttons to control the current process – which also are available in case of some specific applications:



Pause and resume – when pressed for the first time, the currently running process is halted. This happens not immediately but the next time the software is able to perform a halt without causing damage to the current marking process. How fast this halt is done, depends on the used geometries and the capabilities of the used scanner controller card.

When pressed for the second time, a previously halted process is resumed and processing is continued seamless.



Override all stops – this button can not be used to halt a currently running process but to resume it. Different to the standard halt/resume button explained at the left, this also includes stops which have been invoked by process-control plug-ins. This function has to be handled with care: when it is used to override a stop, it may be possible a process is continued while being outside the defined process parameter ranges which itself may end up with an undefined or unusable marking result!

## 5.14 Switching the Laser

Beside the possibility to mark the geometries of the current project, there is also an option to turn on and off the laser for a longer time without moving the scanner. This function is useful e.g. for measuring the laser output or for calibration tasks and can be found in menu "Process" menu item "Toggle laser".

This opens a mark-dialogue-like window which contains a big yellow "Laser"-button to turn on the laser and a red "Stop"-button to turn it off.

**PLEASE NOTE:** when pressing the Start-button, the laser is turned on immediately, so it will fire independent fro many geometry or delays or input signals which may be defined in the current project! So this button switches the laser on directly and without respecting any precondition which may be defined in current project! Thus all security measures have to be active as soon as this dialogue is opened and prior to pressing the Start-button!

Beside start and stop this dialogue offers the following functions:

- - Head – here the head (aka controller card) can be selected for which the laser has to be turned on
- Pen – with this combo box a pen can be chosen. The parameters of this pen (e.g. frequency, power, pulse-width) are used to drive the laser with as soon as the Start-button is pressed
- Edit – by pressing this button the pen settings dialogue is opened which allows it to modify the currently selected pen and resulting from this the parameters the laser is driven with when turned on via this dialogue

## 5.15 Exporting Data

To use the created data on the target system for processing work pieces they have to be stored in a usable format. Here following possibilities exist:

BeamConstruct “.BEAMP” project files:

This is the project file format of BeamConstruct, here all information about used elements and sub-elements, modified geometry, scanner and laser parameters are contained, no information are lost when this format is used.

CSV format:

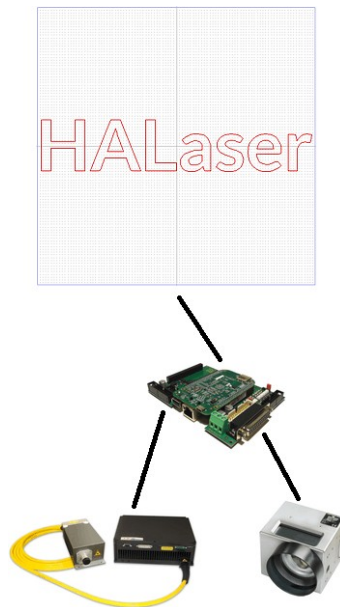
The “comma separated values” format is more a simple list of parameters that are comma-separated. This format includes all geometry and some pen information. Here no dynamic information are contained, only static geometry can be marked when this format is used.

HPGL format:

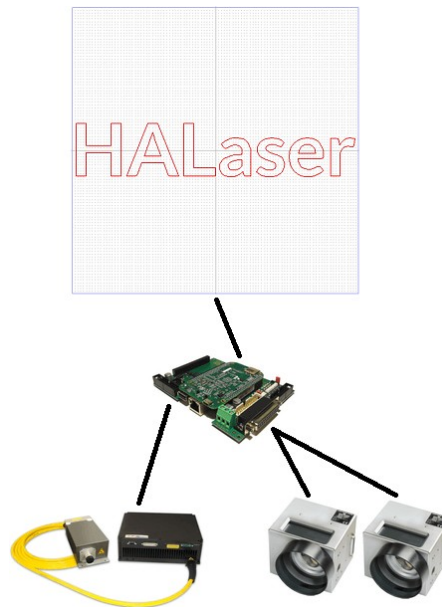
This is a vector graphics format that contains only geometry information, the pen (and thus the processing data) are not part of this format. So it should be used for data exchange with other applications only. This format can be used with the HPGL to Control plug-in. Because of the missing pen information they have to be entered within the plug-ins settings manually. Here no dynamic information are contained, only static geometry can be marked when this format is used.

## 5.16 Multihead Operations

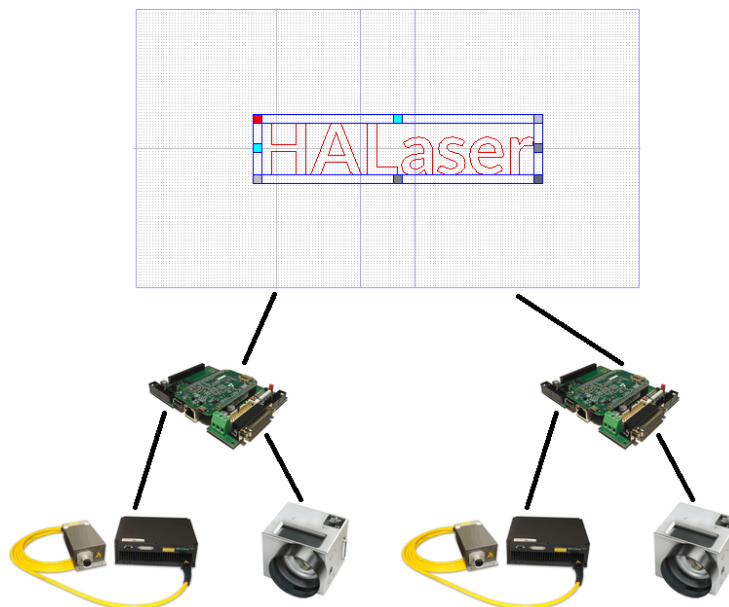
In many cases BeamConstruct will use one scanner card that uses exactly one rectangular working area. In this configuration there will be one scan head performing movements for this working area:



In some special cases this so called singlehead operation mode can make use of a second (and optionally third, fourth,...) working area and a second scan head where both heads perform the same movement. This mode is often called “secondary head”, it simply doubles the output but does not allow to mark different, independent geometries with these two heads:



Full independence of all used heads is possible only with a real multihead mode. In this mode BeamConstruct knows about more working areas that can overlap each other or that can exist beside each other. But all these working areas are completely independent, every area is assigned to an own scan head that performs separate movements:



This multihead mode can be used to:

- mark geometries that are larger than a single working area (in this case it is recommended to let the working areas overlap a bit to profit from vector data optimisations BeamConstruct performs automatically)
- mark large, complex geometries faster (in this case big overlapping areas have to be configured to let both heads work on same area in parallel)
- mark independent geometries faster (in this case working areas do not need to overlap each other)

In last case it is recommended to use scan heads with the same performance and lasers with same power, elsewhere it may happen one marking area slows down the full marking process.

In first two cases different lasers and scan heads can be used but it is recommended to have the more powerful lasers and the faster scan heads configured at the first position.

Setting up of a multihead configuration is quite easy and can be done in general project settings (please refer to detailed description of settings above). There in tab-pane "Hardware" two combo boxes can be found. One is for selecting a scanner controller and one for selecting a scan head. Both can be separate things but don't have to, this depends on the used scanner controller / the used scanner controller plug-in. The difference between both:

- the scanner controller is a piece of hardware / a plug-in that accesses this scanner controller device, where more than one scan head can be connected to.
- the scan head is assigned to a single working area and a single laser that is used for this working area, it is possible to have more than one scan head connected to a scanner controller.

In many cases there will be a 1:1 relation where scanner controller selection is equal to scan head selection, but in case one scanner controller plug-in is able to control more scan heads independently, this separation between "controller" and "head" is necessary.

The same can be found in pen settings (please refer to description above). There the scan head can be selected, means different pen parameters can be set for every working area – independent from the fact which scanner controller plug-in really controls this head.

To configure a multihead-environment, following steps have to be done in global settings of BeamConstruct:

- go to scanner-settings
- in upper combobox select the first scanhead (which is default)
- below of it choose a scanner controller card and configure it, optionally change the field left and top position within the scanner controller card's settings
- select the second scanhead in the combobox on top of the scanner settings tab-pane in global settings
- choose a scanner controller card and configure it → this step now creates a new working area which is assigned to this second scanner controller card and the laser and scanhead it is connected with, so in configuration of this scanner controller the field left and upper size have to be set to appropriate values in order to overlap or to not to be connected with the working field from the first scanner controller card; as soon as a second scanhead/scanner card is configured, BeamConstruct is in multihead-mode automatically
- optionally continue this procedure with a third, fourth, fifth,... scanner controller card and related lasers/scanheads/working fields
- optionally go to tab-pane "Multihead" within project configuration dialogue to configure a specific, preferred behaviour for each of the scanheads created in previous step

## 5.17 IOSelect Mode

Normally a project can be loaded by using the related menu item and by selecting the path and name of the project file manually. This can be automated, using IOSelect mode it is possible to read the state of digital input port of a connected scanner controller and to use the value received from there to automatically load a project file. When this mode is enabled, marking dialogue is opened and the state of the used digital input port changes, a new project is loaded and can be processed during next marking operation. In case the input state changes when a marking process is still running, loading of the new project is performed as soon as the current marking cycle was finished. To set-up IOSelect mode following steps are necessary:

1. create a subdirectory that will hold all the project files
2. create the project files that have to be loaded, here the filename has to consist of a number followed by the file extension ".beamp". The number has to correspond to the bitpattern detected on digital input. So when bit 0 of an input is set, a file "1.beamp" would be loaded. When all bits of an 8-bit digital input are set, a file "255.beamp" would be loaded. When no bits of a digital input are set, a file "0.beamp" would be loaded - nevertheless it is not recommended to use this file name in order to

have a neutral state with no active marking data. All these files have to be placed into directory created in previous step.

3. go to project settings, enable IOSelect-mode in "Misc" panel, specify the directory created above and select the digital input that has to be used for reading the input state. This step requires a scanner controller card configured properly.
4. Select menu "Process" -> "Mark" to activate automatic IOSelect and to start marking operations

Following things have to be noticed:

- when the digital input provides an input pattern for a file name where no corresponding project file exists, current project is flushed so that there are no marking data available
- loading a project automatically does not overwrite the current hardware settings, so all project parameters found in loaded file are ignored and current settings are kept
- in IOSelect mode marking is not controlled by external trigger automatically, to do that all the numbered projects stored in IOSelect-directory have to start with an External Trigger Primary Element and mark-loop option has to be set in Mark-dialogue
- in multihead-environments currently only card 1 can be used to select a project via digital inputs
- IOSelect-mode is started only after mark-dialogue was opened; during switch to an other project mark-dialogue is closed and re-opens automatically afterwards

## 5.18 Command line options

BeamConstruct either can be started as is, then it opens the default view. Beside of that it is possible to hand over some parameters which all are optional. For Windows such a call with additional command line parameters looks like this:

```
BeamConstruct.exe f:/path/to/project.beamp -o mode
```

For Linux such a call would look like this:

```
BeamConstruct.exe /path/to/project.beamp -o mode
```

When a path to a valid BEAMP-project file is handed over, this project is loaded on start-up and shown as default project file.

When the optional parameter `-o` is handed over, a specific working mode can be defined which influences what operation(s) BeamConstruct performs directly after start-up. Here mode can have the following values:

1 – this mode requires a project file to be loaded, it immediately starts a single marking operation with this file and then exits the complete application

2 – when this mode is selected, the “Mark repeatedly”-checkbox in Mark-dialogue is set and disabled so that a user can perform only looped operations and no single marking cycles. Such a looped operation then needs to be interrupted by pressing the Stop-button

3 – in this mode the Mark-dialogue is opened after start-up automatically and the “Mark repeatedly” as well as the “Wait for external trigger” checkboxes are set. For this mode it also is recommended to define a project file to be loaded automatically, elsewhere the user would not have anything to mark



## 5.19 Customise BeamConstruct

### 5.19.1 Rebranding

BeamConstruct can be modified in a way so that it can be delivered together with machinery or additional equipment using an own brand. Here following attributes can be changed:

- the applications name, this value replaces the name BeamConstruct completely
- the homepage which is shown in “About”-dialogue
- the splash image shown during application start-up and in “About”-dialogue
- the application icon shown in taskbar and main window (not supported on all operating systems)
- the copyright as shown in “About”-dialogue; this custom copyright does not replace the default one but is added to it
- the images shown in toolbar and entity-tree
- the menu items which are visible to the end user

To customise the application, different things can be done, which are optional each:

A plain text file with the name `beam.oem` has to be created in UTF-8 format. To become active, it has to be placed in installation directory of the application for Windows operating systems (typically `C:\Program Files\HALsoftware`) or in `/usr/local/openapc` for Linux systems. Within this file every single line contains one customisation definition:

1. the name of the application
2. the homepage URL to be shown in “About”-dialogue
3. path to the splash image in PNG format
4. path to the icon image in PNG format
5. copyright text to be shown in “About”-dialogue
6. optional: RGB hexadecimal values to set an other background colour for the main windows toolbar

As an example:

```
Beamy Raymanipulator
http://www.beamconstruct.com
beamoem.png
beamoem_icon.png
Copyright (c) 2012 by LaserFreak
6060FF
```

This file (in UTF-8 encoding) placed within the directory described above renames BeamConstruct to “Beamy Raymanipulator”, shows <http://www.beamconstruct.com> within the “About”-dialogue (instead of <https://halaser.eu>) as well as the additional copyright information. As splash image for the startup-screen the image out of file `beamoem.png` is used. When this files name is given without an absolute path, the application tries to find it in installation directory (Windows) or in directory `/usr/local/openapc` (Linux). The same is true for `beamoem_icon.png` which is used as image for the applications icon.

### 5.19.2 In-application images

The images that are shown within toolbar and entity tree can be replaced by own ones too. Here two different methods have to be used to replace all of them.

To replace the toolbar- and entity-tree images, alternative images in PNG-format have to be put into the plug-ins location but with different names. The base name is always the name of the plug-in without extension. For the toolbar image a “1.png” has to be added to this base name, for the entity-tree image a “2.png” has to

be added. As an example: The images of the barcode plug-in have to be replaced. It can be found in folder "priplugins" with the file name "libio\_pri\_barcode.dll" or "libio\_pri\_barcode.so". Here two images "libio\_pri\_barcode1.png" with a size of 23x23 pixels and "libio\_pri\_barcode2.png" with a size of 15x15 pixels have to be put into the same folder. After next software start the barcode appears with the new images.

To replace all other images PNG files with special names have to be placed in folder "icons":

- "newprj.png" for the "New"-symbol (23x23 pixels)
- "openprj.png" for the "Load project"-symbol (23x23 pixels)
- "saveprj.png" for the "Save project"-symbol (23x23 pixels)
- "saveprjas.png" for the "Save project as"-symbol (23x23 pixels)
- "quit.png" for the "Quit application"-symbol (23x23 pixels)
- "zoomin.png" for the "Zoom in"-symbol (23x23 pixels)
- "zoomout.png" for the "Zoom out"-symbol (23x23 pixels)
- "zoomworkarea.png" for the "Zoom full editing area"-symbol (23x23 pixels, for compatibility reasons the name of the image contains "work" but not "editing")
- "zoomselected.png" for the "Zoom selected entity"-symbol (23x23 pixels)
- "pointdrag.png" for the "Point drag mode"-symbol (23x23 pixels)
- "drawdir.png" for the "Show marking direction"-symbol (23x23 pixels)
- "openmarkdlg.png" for the symbol to open the mark dialogue (23x23 pixels)
  
- "undo.png" for the "Undo"-symbol in right hand side, vertical toolbar (25x25 pixels)
- "redo.png" for the "Redo"-symbol in right hand side, vertical toolbar (25x25 pixels)
- "group.png" for the "Group"-symbol in right hand side, vertical toolbar (25x25 pixels)
- "ungroup.png" for the "UnGroup"-symbol in right hand side, vertical toolbar (25x25 pixels)
- "alignleft.png" for the "Align left"-symbol in right hand side, vertical toolbar (25x25 pixels)
- "centrehoriz.png" for the "Centre horizontal"-symbol in right hand side, vertical toolbar (25x25 pixels)
- "mirrorx.png" for the "Mirror horizontal"-symbol in right hand side, vertical toolbar (25x25 pixels)
- "mirrory.png" for the "Mirror vertical"-symbol in right hand side, vertical toolbar (25x25 pixels)
- "alignright.png" for the "Align right"-symbol in right hand side, vertical toolbar (25x25 pixels)
- "aligntop.png" for the "Align top"-symbol in right hand side, vertical toolbar (25x25 pixels)
- "centrevert.png" for the "Centre vertical"-symbol in right hand side, vertical toolbar (25x25 pixels)
- "alignbot.png" for the "Align bottom"-symbol in right hand side, vertical toolbar (25x25 pixels)
- "booladd.png" for the "Boolean add"-symbol in right hand side, vertical toolbar (25x25 pixels)
- "boolsub.png" for the "Boolean subtract"-symbol in right hand side, vertical toolbar (25x25 pixels)
- "boolinter.png" for the "Boolean intersect"-symbol in right hand side, vertical toolbar (25x25 pixels)
  
- "dlgmark\_pilot.png" for the pilot laser symbol in mark dialogue (113x113 pixels)
- "dlgmark\_mark.png" for the mark-start symbol in mark dialogue (113x113 pixels)
- "dlgmark\_stop.png" for the stop button symbol in mark dialogue (113x113 pixels)
  
- "penwizard.png" for the pen parameter wizard button in pen settings dialogue (55x55 pixels)

- “rotate\_move\_left\_xpm.png”, “rotate\_step\_left\_xpm.png”, “rotate\_step\_right\_xpm.png”, “rotate\_move\_right\_xpm.png” and “rotate\_to.png” for rotational movement buttons in “Motion” tab-pane and mark dialogue (17x17 pixels)
- “arrow\_move\_down\_xpm.png”, “arrow\_step\_down\_xpm.png”, “arrow\_step\_up\_xpm.png”, “arrow\_move\_up\_xpm.png” and “vertical\_to.png” for planar, vertically oriented movement buttons in “Motion” tab-pane and mark dialogue (17x17 pixels)
- “arrow\_move\_out\_xpm.png”, “arrow\_step\_out\_xpm.png”, “arrow\_step\_in\_xpm.png”, “arrow\_move\_in\_xpm.png” and “longit\_to.png” for planar, longitudinally oriented movement buttons in “Motion” tab-pane and mark dialogue (17x17 pixels)
- “arrow\_move\_left\_xpm.png”, “arrow\_step\_left\_xpm.png”, “arrow\_step\_right\_xpm.png”, “arrow\_move\_right\_xpm.png” and “horiz\_to.png” for planar, horizontally oriented movement buttons in “Motion” tab-pane and mark dialogue (17x17 pixels)

### 5.19.3 Custom menus and functions

To hide specific menu items from BeamConstruct main window, a plain text file with the name `beammenu.oem` has to be created. To become active, it has to be placed in installation directory of the application for Windows operating systems (typically `C:\Program Files\HALsoftware`) or in `/usr/local/openapc` for Linux systems. This text file contains one number for each menu item which has to be disabled. Every number has to be placed in a new line. Following numbers exist and disable the menu items as described below (lists are ordered in appearance of menu items).

For menu “Project”:

- 1001 – New project
- 1002 – Open project
- 1013 – Insert Project...
- 1014 – Save hardware settings...
- 1015 – Save payload data...
- 1016 – Load hardware settings...
- 1017 – Load payload data...
- 1018 – Recent projects
- 1003 – Import...
- 1004 – Save project
- 1005 – Save project as...
- 1010 – Save with options...
- 1101 – Export CSV
- 1102 – Export HPGL
- 1103 – Export CLI
- 1104 – Export RWX
- 1006 – Project settings...
- 1007 – Pen settings...
- 1008 – Save as default configuration
- 1009 – Load default configuration
- 1018 – Machine settings...

For menu “Edit” and the related toolbar-items of the right, vertical toolbar:

- 2101 – Group
- 2102 – Group to Active Split
- 2103 – Group to Active Move
- 2104 – UnGroup
- 2105 – Duplicate
- 2106 – Align left
- 2107 – Centre horizontally
- 2108 – Align right
- 2109 – Align top
- 2110 – Centre vertically
- 2111 – Align bottom
- 2112 – Group to Active Rotational Split

- 2113 – Reverse direction
- 2201 – Merge
- 2202 – Split
- 2203 – Optimise
- 2204 – Reduce Geometry
- 2205 – Extrude
- 2206 – Delete Element(s)
- 2207 – Boolean add
- 2208 – Boolean subtract
- 2209 – Boolean intersect
- 2210 – Project to ground
- 2308 – Start Video
- 2301 – Freeze Video
- 2302 – Calibrate Camera
- 2309 – Crop Camera
- 2307 – Drop Calibration
- 2303 – Teach Fiducial
- 2304 – Load Fiducial
- 2305 – Save Fiducial
- 2306 – Drop Fiducial
- 2001 – Undo
- 2002 – Redo
- 2901 – the complete “Vision” submenu including “False colours”
- 2902 – the complete “False colours” submenu

For menu “Process”:

- 3001 – Simulate
- 3002 – Mark
- 3017 – Toggle laser
- 3005 – Write Stand Alone Data
- 3009 – Send Stand Alone Data
- 3010 – Send Named Stand Alone Data
- 3013 – Show State
- 3006 – Increment
- 3007 – Decrement
- 3008 – Reset

For menu “Help”:

- 4002 – License
- 4003 – Reset License
- 4006 – Generate bugreport
- 4007 – Adjust projection

For the plug-ins within the upper, horizontal toolbar:

- 10001 – Curve Postprocessing Element
- 10002 – Sine Postprocessing Element
- 11001 – Barcode Primary Element
- 11002 – Bezier Primary Element
- 11003 – Circle Primary Element
- 11004 – Delay Primary Element
- 11005 – Dot Primary Element
- 11006 – External Trigger Primary Element
- 11007 – Custom Input Primary Element
- 11008 – Custom Output Primary Element
- 11009 – Single Line Primary Element
- 11010 – Motion Primary Element
- 11011 – Polygon Primary Element
- 11012 – Rectangle Primary Element

- 11013 – Serial Port Output Primary Element
- 11014 – Spiral Primary Element
- 11015 – Star Primary Element
- 11016 – Text Primary Element
- 11017 – Triangle Primary Element
- 11018 – Z-Shifter Primary Element
- 12001 – Hatch Additional Element
- 13001 – CSV File Data Input Element
- 13002 – OAPC Data Input Element
- 13003 – Serial/Date/Time Input Element

As an example: a file `beammenu.oem` with following contents:

```
1013
2102
2103
3001
```

would hide the following menu items from the main window so that they can't be used any longer:

- menu "Project", menu item "Insert project..."
- menu "Edit", menu items "Group to Active Split" and "Group to Active Move"
- menu "Process", menu item "Simulate"

#### 5.19.4 Predefined hardware settings

Normally the hardware settings are provided with the project file which typically contains all required definitions. As there are machines possible which require different settings (e.g. because of exchangeable optics), there is an easy way to give the user access to a list of predefined settings. For this following steps have to be done:

- create an own directory "presets". On Windows operating system this directory has to be created in installation folder of the software (means right beside the BeamConstruct.exe). For Linux operating systems the presets-folder has to be created in `/usr/share/openapc/`
- place a set of .BEAMH hardware settings in this directory which represent the different hardware configurations you want to offer to the user. These .BEAMH files can be created via menu "Project" → "Save hardware settings...". The file name of the .BEAMH file later is used as name for the menu entry.
- restart BeamConstruct. Now in menu "Project" a new sub-menu "Load preset" appears which itself contains all the .BEAMH-filenames as sub-menuitems itself. Now a hardware configuration can be chosen simply by selecting one of these menu items.  
PLEASE NOTE: as long as the "presets"-directory exists, hardware configurations that come with a .BEAMP project file are ignored completely! Instead of that always the hardware settings out of the currently chosen preset are used.

#### 5.19.5 Custom geometries

By default BeamConstruct comes with a set of geometric elements which can be accessed via the toolbar. Here all base elements such as circle, rectangle, triangle, text, barcode and others are contained. These base elements can be extended by own, custom geometries which also are accessible via the toolbar. When set up properly, the elements appear as sub-tool within a new toolbar entry. Once the related element was

selected, it can be placed as static geometry (which can be hatched, scaled rotated,... like any other geometry) within the drawing view.

To set up such a custom geometry tool, following steps have to be performed

1. draw the desired geometry within the drawing view
2. place the whole geometry in a proper way around the coordinate position 0,0; this coordinate position will be the centre point of the geometry later
3. go to menu "Export as" → "HPGL" and save the geometry within the installation directory of the main application and within a subfolder "templates" (so for Windows the location to save the HPGL-file to probably would be "C:\Program Files\HALsoftware\templates\"). The file name of the saved HPGL file should be an English name describing the geometry, the extension has to be .plt
4. create a 25x25 pixel wide PNG file that symbolises the custom geometry and save it in same templates-directory using the same base name and the extension .plt
5. Optionally: translate the name of the new toolbar element using the custom\_xx.property translation file (as described in SDK-manual from [https://sourceforge.net/p/oapc/code/ci/master/tree/sdk\\_manual.pdf](https://sourceforge.net/p/oapc/code/ci/master/tree/sdk_manual.pdf) section "Localisation")

Now when you restart the application, you find a new toolbar-element which shows a pen. When clicking this pen-symbol, a menu opens where the custom geometry is shown (using the PNG image from the last pen above). After clicking this symbol and after left-clicking within the drawing view, the related geometries are placed within the drawing view at the point where the left-click was done. Here for the new geometry the clicked position is similar to position coordinate 0,0 out of step 2) above.

## 5.19.6 Adding Languages

### 5.19.6.1 Working Principle

BeamConstruct uses a translation and localisation concept that is very simple and bases on plain text files that can be edited easily. The selection of the correct language file is done by the application automatically, here the default language of the underlying operation system is used. So no user interaction or configuration is required. Alternatively it is also possible for the user to set the language to be used via the machine settings (as described in "5.6 Machine Configuration").

### 5.19.6.2 Choosing the Correct Translation File Name

First of all the names of the files that contain the translations have to be chosen. The base file names are

- **openapc\_xx\_YY.property** or **openapc\_xx.property** – for basic texts that are spread via plug-ins and applications
- **construct\_xx\_YY.property** or **construct\_xx.property** – for the texts that are used within BeamConstruct only
- **common\_xx\_YY.property** or **common\_xx.property** – general texts that are used by all applications
- **custom\_xx\_YY.property** or **custom\_xx.property** – an additional translations-file that can be used for own translations (e.g. texts that are defined in user interface elements); this file does not exist by default and therefore will not be overwritten during installation of software updates

Here "xx" and "YY" specify the language: "xx" is the general language (according to ISO 639-1, compare to [http://en.wikipedia.org/wiki/List\\_of\\_ISO\\_639-1\\_codes](http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes)) and "YY" describes the country code (according to ISO 3166, compare to [http://www.iso.org/iso/english\\_country\\_names\\_and\\_code\\_elements](http://www.iso.org/iso/english_country_names_and_code_elements)). In most cases the "YY" part can be dropped.

An example: when BeamConstruct is started on an English system that is located in United Kingdom, it tries to load the translation files **openapc\_en\_GB.property**, **common\_en\_GB.property** and **construct\_en\_GB.property**. If they can't be found, within a next step it tries to find general English language files **openapc\_en.property**, **common\_en.property** and **construct\_en.property**. If that fails too it falls back to the internal default language.

So you are open to use both names with the language code and the country code or only with the language code. In latter case you cover more language variants but you are not able to take care about regional and country-related peculiarities.

The files itself have to be deployed within the sub-folder “translations” of the applications installation location. In case a file can be found there at start up, it is used by the applications automatically.

### 5.19.6.3 *Translating the Applications*

The contents of a language file are quite simple: it is a plain text file in UTF-8 encoding that contains pairs of values that are separated by “ = ” (space, equal-sign, space). On the left side of these separators the original, default texts can be found, they can't be edited and never should be changed. On the right side you can place your translation. So following rules apply to such a language file in order to create a valid translation for a language:

- the file has to be stored with UTF-8 encoding
- the original texts on the left side never have to be changed
- the delimiter between original and translated text has to be a combination “space, equal sign, space” (“ = ”)
- translations can't be done using different lines, they have to be placed in one line directly after the original text and the delimiter characters
- you can add a “\n” to get a line break wherever the original text makes use of such a line break too (Example: a text “this is\nthe second line” would result in an output

```
this is
the second line
```

only when the original text also makes use of the “\n” symbol)

An example to make it more clear:

```
My original text = Mi texto original
This is my\noriginal text = Este es mi\ntexto original
```

Here the original texts stay on the left side, within the second line the special “\n” symbol is used. That means, that also the translation can make use of a line break here. In both cases the text on the left side is specified by the application and has not to be modified. On the right side the translated text is placed (this example uses a language where we think it could be Spanish). So for such an example the language file could have e.g. the name **openapc\_es.property**.

After the original texts are fixed and defined by the contents of the application, you have to use an existing translation. Copy that file, set the correct file name for it and extend its contents by your translation – that's all!

## 5.20 **BeamLock**

BeamLock is a tool that can be used to apply some own lock-rules to a system. This might be useful e.g. for machine integrators that ship hardware with installed BeamConstruct to customers before the full invoice is paid. In this case a lock can be placed on such a system that becomes active after a given time. This means

the machines functionality is time-limited. Now the user would have to enter a password that is provided by the vendor of the machine only when invoice is paid/licensing conditions are met/agreements are redeemed. Without the password the software installation is no longer usable, also deinstalling and installing BeamConstruct again will not solve this issue.

BeamLock provides the possibility to define several time ranges with own passwords for each of these time ranges. Thus it is possible to define several lock grades that become active after different time ranges. Additionally a main password has to be defined which can be used to override the time-dependent locks at once and to disable the full lock immediately. Beside of that it is also possible to dynamically create a password which incorporates a time range and gives the possibility to expand a given lock-time again and again without the need to use the predefined lock-ranges.

PLEASE NOTE: all passwords including the main passwords have to be completely different in order to have a good security. Beside of that it is recommended to use complex passwords with a length of at least 8 characters and with upper- and lowercase characters that itself do not form meaningful words. Examples for BAD PASSWORDS are: "mylock1", "mylock2", "mylock2" and "mylockm". Examples for good passwords are: "XdlfTr238", "yERf12rt67", "934trAwX4", "m4RT56ZZtgd34".

### 5.20.1 Fixed time range locks

To apply different locks with several fixed time ranges on a system, following steps have to be performed:

1. Start BeamLock on the system that has to be locked. For Windows BeamLock.exe can be found in HALsoftware installation directory, for Linux it has to be started out of a console by entering command "BeamLock". On none of the supported systems BeamLock comes with an icon in start menu.
2. Set the "Enable" checkbox for every row where a time range and a password has to be provided; specify the time after the lock has to become active and a password for unlocking it.
3. Specify a main password.
4. Press the button "Lock System". When a system is already locked you have to specify the current main password in order to override this lock. When a system was not locked before but contains a fresh and clean installation, it may be the software asks for a password too: in this case use "HALaser" as main password. When "HALaser" (without the quotation signs) does not work for you, your system was already locked and you cannot overwrite the current lock without knowing the main password that was used during this lock. In such a case you have to contact the vendor of the system, there is no way to remove such a lock when the main password is not known.
5. Write down your passwords and keep them secure. PLEASE NOTE: these passwords and especially the main password are the ONLY way to access your system after specified time has elapsed or to set a new lock! Once again: There is no way to remove such a lock when the main password is not known, so keep these passwords secure and do not press the "Close" or "Cancel" button before you got them!

PLEASE NOTE: For Windows locking procedure it depends on the privileges BeamConstruct is executed with by default. When it is running using an Administrator account the steps described above have to be done using BeamLock.exe together with the same Administrator account. When it is executed using a non-privileged user account the steps described above in every case have to be done twice using BeamLock.exe: once using this users account and once using a fully privileged Administrator account! When doing these steps two times, it is recommended to use the same passwords in both cases to avoid confusion when end user plays around with the used installation and does changes in environment BeamConstruct is used.

PLEASE NOTE: For Linux it is assumed BeamConstruct is not executed as root but as a non-privileged user. In this case the steps described above in every case have to be done twice using BeamLock: once using this users account and once using the root-account! When doing these steps two times, it is recommended to use the same passwords in both cases to avoid confusion when end user plays around with the used installation and does changes in environment BeamConstruct is used.



## 5.20.2 Dynamic time range lock

Beside the possibility to define one or more fixed time ranges before shipping of a system, it is also possible to generate new passwords which extend the lock time again and again. When such a password is sent to a user, a current lock can be removed for the time which is defined by this password. For creating such a dynamic time range lock, following conditions have to be met:

- the target system needs to be locked by the steps described in previous section and the time period of such a lock needs to have elapsed but the system is not yet unlocked by using one of the predefined passwords
- the main password of this system is known

Comparing to what is described in previous section, such a lock can be created also when the target system is no longer available:

1. Start BeamLock on any available system. For Windows BeamLock.exe can be found in HALsoftware installation directory, for Linux it has to be started out of a console by entering command "BeamLock". On none of the supported systems BeamLock comes with an icon in start menu.
2. Enter the main-password in related input field and confirm it by pressing "Enter"
3. In lower row of input fields right below of the main password, specify the time a system has to be unlocked for and press button "Create"
4. Now a dialogue opens where the unlock password which belongs to this main password and the given time is shown and where it can be copied out of the text field directly.

## 5.21 BeamServer Remote Control Interface

### 5.21.1 Overview

BeamConstruct provides an operation mode where whole application can be remote-controlled. This can be used to do some script-or program-controlled automation or to hide BeamConstruct while (visually) performing some other tasks in foreground. To do this, a special program is provided: BeamServer. When it is started (BeamServer.exe under Windows, BeamServer when using Linux) instead of BeamConstruct, a TCP server socket is created using port 11350 (or optionally any other available port). Now an external application can connect to this socket and send some specific ASCII-commands. These commands then invoke usage of BeamConstruct itself. To make that clear: BeamConstruct does not need to be started in this case, this will be done by BeamServer automatically when related commands are sent.

### 5.21.2 Usage

A standard sequence of commands sent to BeamServer always looks like this:

1. Start BeamServer with a number as command line parameter specifying which parts of the UI have to be shown
2. Create a TCP-connection to the IP of the computer where HAL software package is installed and where BeamServer is running at, connection has to be established to port 11350 (this port can be changed using command line parameter -p)
3. and following: send several ASCII-commands as described below to control BeamConstruct

finally: Send

ExitUI

to shut down and close BeamConstruct and BeamServer

When TCP connection to BeamServer is interrupted somewhere within this sequence, BeamConstruct is NOT shut down. After re-connecting to BeamServer a controlling application can continue at the point where it has been before interruption. Beside of that only one connection to BeamServer is allowed at the same time.

For testing purposes it is also possible to connect with a normal Telnet-client to BeamServer and to give commands manually. In this case Telnet client has to be used in RAW operational mode (auto-negotiation deactivated).

### 5.21.3 Starting BeamServer

BeamServer expects a command line parameter specifying which parts of BeamConstruct UI have to be shown. This parameter is a number. When this number is set to 0, BeamConstruct acts completely invisible. When it is set to 255, full GUI is shown. Between these numbers it is possible to define some subsets of the GUI to be shown just by adding the following values and giving the result as parameter:

- 1 – show drawing area (the big area in the middle where vector data are drawn)
- 2 – show toolbars (on top of main window where some basic functions and elements are accessible)
- 4 – show left hand side element panels
- 8 – show right hand side element tree
- 16 – show the menu bar with all menu items
- 32 – show the close button in frame of main window
- 64 – show the status bar at bottom of main window
- 131072 – disables the possibility for the user to exit the application; when this value is enabled, the application can be left via command "ExitUI" only
- 262144 – disable all possibilities for a user to edit projects or geometries, when this value is set, even dragging, rotating or scaling of elements is no longer possible
- 524288 – the warning-dialogue on start-up is disabled; this value should be used only very rarely and only when user is already informed what the software does so that there can't be any surprising and therefore dangerous operations
- 1048576 – applies only to mark-dialogue, when this value is used, the manual scanner control tab-pane in mark-dialogue is not shown
- 2097152 - applies only to mark-dialogue, this hides the part counter tab-pane
- 4194304 – applies only to mark-dialogue, it hides the "Motion" tab-pane
- 8388608 – applies only to mark-dialogue, when this value is set, the "Geometry" tab-pane in mark-dialogue is hidden
- 16777216 – applies only to mark-dialogue, this hides the pilot laser tab-pane
- 33554432 – applies only when value 2 (see above) is used and when value 1073741824 (see below) is not set, this option hides the "edit pen" button in "Element" tab pane
- 67108864 – this centres and locks drawing area (in case it is visible by setting the value 1) so that its positions can't be changed by dragging the mouse; the view can be changed only by sending commands that modify the current zoom
- 134217728 – applies only when value 2 (see above) is used, it inhibits the possibility to edit single points; when this value is used, the coordinates grid in "Geometry" tab-pane is disabled; it is not necessary to set this value when 262144 (see above) is already set
- 1073741824 – applies only when value 2 (see above) is used, if it is set, the "Element" panel is not shown
- 536870912 – applies only when value 2 (see above) is used, this hides the dynamically changing panel with the parameters of the currently selected element
- 1073741824 – applies only when value 2 (see above) is used, this disables and hides the layers-panel

2147483648 – do not load and use locally stored default settings, the software starts with an (empty) default configuration when this initialisation value is used

As second, optional command line parameter option “-p” can be given together with a number in range 1024..65535. This number specifies the port the BeamServer has to listen at.

As an example: "BeamServer.exe -p 1200 17" would show BeamConstruct consisting of drawing area (=1) and menu bar (=16) only and would bind the server to port 1200 (instead of 11350).

For Windows please note: BeamServer.exe has to be started using HALsoftware installation directory as working directory. To ensure this directory is used, it is recommended to create a shortcut to BeamServer.exe out of the HALsoftware installation directory, this configures the correct directory automatically. Additionally it is possible to set the parameter within the shortcut definitions too. When BeamServer.exe is started using any other directory, it will not work properly and will not be able to use any of the elements and plug-ins.

#### 5.21.4 Remote Control Commands

Every command sent to the BeamServer has to be terminated with a carriage return/line-feed (CR/LF, \r\n, exact implementation depends on the used programming language/interface). Each command that is sent to BeamServer via TCP/IP is responded either by a text "OK" when operation was successful or by a text "ERROR" signalling the last command could not be executed successfully. When there is no such response, the command has not arrived at the BeamServer.

"ERROR" is always followed by two error codes where the first one describes the kind of the error that happened and the second one is an internal error number that may be helpful in case of problems of the BeamServer itself. For first error code following values are possible:

3 – OAPC\_ERROR\_DEVICE – tried to access an external device (scanner controller card, motor controller,...) which was configured for usage but which was not accessible

5 – OAPC\_NO\_DATA\_AVAILABLE – some elements or data a command is referring to are not available or could not be found

7 – OAPC\_ERROR\_RESOURCE – tried to access a resource that does not exist (e.g. when using UID or name of an element that could not be found)

11 – OAPC\_ERROR\_NO\_MEMORY – not enough memory available

12 – OAPC\_ERROR\_STILL\_IN\_PROGRESS – function could not be executed because an other operation is still in progress which has to be completed prior usage of this function

16 – OAPC\_ERROR\_INVALID\_INPUT – the data/parameters given within last command are incomplete or wrong

18 – OAPC\_ERROR\_CREATE\_FILE\_FAILED – creation of a new file/appending data to an existing file failed, file could not be opened

19 – OAPC\_ERROR\_OPEN\_FILE\_FAILED – opening a file for reading failed (happens e.g. when the file does not exist or is not accessible for some reason)

20 – OAPC\_ERROR\_WRITE\_FILE\_FAILED – writing data into an already opened file failed (e.g. because there is no more space left on a device)

21 – OAPC\_ERROR\_READ\_FILE\_FAILED – reading data from an already opened file failed either because the file is damaged or truncated or because the file format is not as expected

24 – OAPC\_ERROR\_INTERNAL – this error points to a software-internal problem; when it occurs, please create a bugreport (as described in “5.4.2 The Help menu”) and contact us!

Following commands are supported by BeamServer:

#### **ExitUI**

After sending this command BeamConstruct is stopped and - if visible - the GUI of it is closed. The BeamServer and it's TCP/IP socket is shut down as well. Afterwards it is not possible to send any other command.

#### **CreateLog <path>**

Creates a (new) logfile at the given path. This logfile will contain all commands sent to the BeamServer, all responses on these commands sent back to the client which is connected to the BeamServer and all notes generated with command `WriteLogNote`. When this command is called and a logfile is already open and used, the old one is closed and log data are written to the new file. When the file specified by parameter `path` already exists, the contents of this logfile are deleted and a new file is created.

Every line within the logfile starts with a fixed character to identify the type of log entry:

C: - command sent from client to BeamServer

R: - response to a command, sent from BeamServer back to client

N: - log note as set by command `WriteLogNote`

#### **WriteLogNote <text>**

Writes the given text into a logfile. To use this command, a log has to be created by calling `CreateLog` before. The `text` specified as parameter to this command is written to the log file without any further modifications.

#### **CmdStopMark**

Tries to stop a running marking operation in the same way like it would be done by a user pressing the "STOP"-button in mark dialogue.

This command is intended to be used with mark-operations which work non-blocking, means where the command returns immediately and not only after operation has finished. For all blocking commands it would not make sense to send `CmdStopMark` since the BeamServer would not be able to receive and to process this command.

#### **CmdIsMarking**

Checks if an asynchronous marking operation is still pending/a mark dialogue is still open. This function returns 0 when no mark operation is pending and 1 when marking is active.

#### **TriggerUI <number>**

This command can be used to invoke any action that would be possible by user interaction within BeamConstruct too. The given `<number>` specifies which operation has to be triggered, here every user action like selecting a menu item or pressing a toolbar is possible. Most of the commands return immediately, independent if the operation was successful or not. Synchronous commands that return only when related operation was finished are mentioned explicitly below. Currently following `<number>`-values are possible:

All four-digit numbers correspond to menu items and/or tool bar buttons and their functionality within the main application, when such a number is used the related operation is invoked (for a more detailed description please refer to BeamConstruct documentation above):

1001 – new project

1002 – load existing project file

1003 – import custom file format (as supported by BeamConstruct: vector and raster image formats)  
 1004 – save the current project using a known name  
 1005 – save the current project using a new name  
 1101 – export vector data in CSV format  
 1102 – export vector data in HPGL/PLT format  
 1103 – export layer vector data in CLI format  
 1104 – export 3D mesh data in RWX format  
 1006 – open general project settings dialogue  
 1007 – open pen settings dialogue  
 1008 – save the current configuration as default settings  
 1009 – load default settings  
 1010 – save project with options  
 1011 – import 3D custom file format (as supported by BeamConstruct: vector formats to be used as 3D mesh)  
 1012 – open general settings dialogue right as code 1006 but brings the “Scanner” tab pane to front  
 1013 – insert a new .BEAMP project file without deleting already existing project data  
 1014 – save hardware settings into a .BEAMH file  
 1015 – save geometries, control elements and pen definitions into a .BEAMV file  
 1016 – load hardware settings from a .BEAMH file and leave geometries, control elements and pen definitions unchanged  
 1017 – load geometries, control elements and pen definitions from a .BEAMV file and leave current hardware settings unchanged  
 1018 – open general machine settings dialogue  
 1019 – import custom file format (as supported by BeamConstruct: vector, 3D and raster image formats) in case of a 3D mesh add the data to an already existing Active Slice Group instead of creating a new, separate one  
 1020 – import custom 3D file format (as supported by BeamConstruct: 3D meshes) and add it to an already existing Active Slice Group instead of creating a new, separate one  
 1021 – import custom bitmap image file format and open the related import dialogue to let the user set import parameters

2001 – undo last operation  
 2002 – redo last undone operation  
 2101 – concatenate all selected elements into one group  
 2102 – concatenate all selected elements into a splitgroup which can perform rotational and 1D/2D planar splits  
 2103 – concatenate all selected elements into a movegroup  
 2104 – ungroup the elements out of the selected group  
 2105 – duplicate the selected element  
 2106 – right-align the selected element  
 2107 – centre the selected element horizontally  
 2108 – left-align the selected element  
 2109 – top-align the selected element  
 2110 – centre the selected element vertically  
 2111 – bottom-align the selected element  
 2112 – concatenate all selected elements into a splitgroup which can perform only rotational splits  
 2113 – reverse the marking direction of the selected element  
 2201 – merge the geometries of the selected element into a static one  
 2202 – split the geometries of the selected element logically  
 2203 – optimise the geometries of the selected element  
 2204 – reduce the geometries of the selected element using a lossy algorithm  
 2205 – extrude the selected element into third dimension  
 2206 – delete the currently selected element(s)  
 2207 – Boolean add  
 2208 – Boolean subtract  
 2209 – Boolean intersect  
 2210 – Project to ground  
 2301 – freeze the currently shown background video  
 2302 – calibrate the currently connected camera  
 2303 – teach fiducials

2304 – load fiducials  
 2305 – save taught fiducial(s) 2306 – drop taught fiducial(s) 2307 – drop camera calibration data 2308 – toggle image capturing on or of (when configured to be started manually in global project settings) 2309 – configure cropping of camera image  
 3001 – start process simulation  
 3002 – opens the marking dialogue to let the user control the marking process  
 3003 – marks the current project once, here the mark dialogue is opened to let the user stop the marking process if necessary; the dialogue is closed automatically as soon as marking was finished; the return information "OK" for this command is sent only after marking process has finished, thus it is not necessary to check the marking state, as soon as a response is received, marking is complete (or was stopped)  
 3004 – marks the current project silently and without opening the mark dialogue; the return information "OK" for this command is sent only after marking process has finished, thus it is not necessary to check the marking state, as soon as a response is received, marking is complete (or was stopped)  
 3006 – increment the elements in current project  
 3007 – decrement the elements in current project  
 3008 – reset the elements in current project to their default/start values  
 3013 – open devices when not already done and show scanner card state dialogue  
 3015 – marks the selected elements out of the current project once, here the mark dialogue is opened to let the user stop the marking process if necessary; the dialogue is closed automatically as soon as marking was finished  
 3017 – open the dialogue to turn the laser on/off manually using a selected pen  
 3020 – this command starts marking of the current project in a loop by using the fully equipped mark dialogue. Marking does not start immediately but requires an external trigger signal. This mark dialogue and mark-loop can be left by pressing the stop-button, by clicking the close-symbol of the mark-dialogue itself or by sending command `CmdStopMark`.  
 3021 – start the pilot laser with the mark dialogue visible, pilot laser mode and number of loops can be set with commands `CmdSetPilotMode` and `CmdSetPilotLoop`  
 3022 – start the pilot laser without a visible mark dialogue, pilot laser mode and number of loops can be set with commands `CmdSetPilotMode` and `CmdSetPilotLoop`  
 3023 – this command is similar to command 3020, it starts marking of the current project in a loop but it does not show any mark dialogue where the operation could be stopped by the user. Marking does not start immediately but requires an external trigger signal. This mark dialogue and mark-loop can be left only by sending command `CmdStopMark`. As long as this looped mark operation is running, the user interface is locked and no manual operations are possible with `BeamConstruct`.  
 3024 – this command starts marking of the current project in a loop like command 3020 but by using a mark dialogue which consists of a stop-button only. Marking does not start immediately but requires an external trigger signal. This mark dialogue and mark-loop can be left by pressing the stop-button, by clicking the close-symbol of the mark-dialogue itself or by sending command `CmdStopMark`.  
 4001 – show the "about"-dialogue  
 4002 – show license information  
 4003 – reset the current license  
 4006 – generate a bugreport  
 4007 – open the dialogue to adjust a preview projection  
 4008 – save a protocol of the current loaded project/used configuration  
 5001 – switch the current drawing area view to (default) top view  
 5002 – switch the current drawing area view to front view  
 5003 – switch the current drawing area view to right side view  
 5004 – switch the current drawing area view to 3D view  
 5005 – switch the current drawing area view to split mode with all views combined

All five-digit numbers correspond to remaining tool bar buttons and their functionality within the main application, please refer to section „BeamConstruct“ above to get detailed information about them:

10001 – zoom into view  
 10002 – zoom out of view  
 10003 – zoom to view full working area

10004 – zoom to fully view selected element  
10005 – enable pointdrag mode for selected element  
10006 – log in a user  
10007 – toggle the mode which shows/hides mark start point and marking direction in geometry  
10008 – zoom to view full editing area

#### **CmdNewPrj**

Clears all current data and flushes all existing elements, settings and pens. After calling this command default settings are restored and project does not contain any geometries. When `CmdAppendPrj` is used afterwards to load a new project, current (default) settings are kept.

#### **CmdClrElems**

Removes all elements of the current project. Comparing to `CmdNewPrj` after a call to this command the current configuration is still available and the connections to used controllers are unchanged. So this function does not switch back to the default configuration but behaves like all available geometries and elements have been deleted separately.

#### **CmdLoadPrj <path>**

Loads a new .BEAMP project from given <path>. The elements contained in the given project are appended to the current ones (if there are any). The current settings and pens are overwritten by the ones found in the loaded project file.

#### **CmdSavePrj <path>**

Saves the current project into a .BEAMP project file as specified by <path>.

#### **CmdLoadPenPrj <path>**

Loads the pens out of a .BEAMP project from given <path>. This project has to contain only pen definitions, no vector data and no hardware settings. When this command is used, all current pens are removed, the pen definitions out of this file are loaded and when it contains less pens than the original project, the missing pens are filled up using the default pen settings out of the loaded file. So this function can be used to replace pen definitions without loading a completely new project.

#### **CmdSavePenPrj <path>**

Saves the pens out of the current project into a .BEAMP project at the given <path>. This project then will contain only pen definitions, but no vector data and no hardware settings.

#### **CmdAppendPrj <path>**

Loads a new .BEAMP project from given <path>. The elements contained in the given project are appended to the current ones (if there are any). The current settings and pens are kept, the ones in loaded project file are ignored.

#### **CmdGetPrjName**

Returns the (file) name of the currently loaded project. When no project is loaded, an empty line is returned.

#### **CmdGetPrjPath**

Returns the full path name of the currently loaded project. When no project is loaded, an empty line is returned.

#### **CmdGetEstMarkTime**

Returns the estimated marking time of the current project in unit milliseconds. When no project is loaded, the returned marking time will be 0.

#### **CmdImport <path>**

Imports some vector data that do not have standard .BEAMP project file format. The imported element then gets a name which consists of path separator and file name. As an example: when a file is imported from C:\tmp\test.svg the imported element will get the name \test.svg

#### **CmdImportSilent <path>**

Imports some vector data that do not have standard .BEAMP project file format. Comparing to CmdImport here importing is done silently and without any configuration dialogues opening during this operation. Instead of this data are imported using default parameter. The imported element then gets a name which consists of path separator and file name. As an example: when a file is imported from C:\tmp\test.svg the imported element will get the name \test.svg

**CmdSetCharIO2 <new text>**  
**CmdSetCharIO3 <new text>**  
**CmdSetCharIO4 <new text>**  
**CmdSetCharIO5 <new text>**

Set a new ASCII text to be used for all input elements that make use of the logical "Char" inputs. When such an input element is assigned to a text or barcode primary element, the text or barcode is replaced by the given <new text> for the next marking operation. The four commands for char IO 2..5 correspond to the inputs 2..5 that can be selected in used input elements. The <new text> has to be given in plain 8 bit ASCII (local default encoding). For changing texts in Unicode-encoding, please refer to functions CmdSetUniIOx below.

**CmdSetUniIO2 <new text>**  
**CmdSetUniIO3 <new text>**  
**CmdSetUniIO4 <new text>**  
**CmdSetUniIO5 <new text>**

Set a new Unicode text to be used for all input elements that make use of the logical "Char" inputs. When such an input element is assigned to a text or barcode primary element, the text or barcode is replaced by the given <new text> for the next marking operation. The four commands for char IO 2..5 correspond to the inputs 2..5 that can be selected in used input elements. The <new text> has to be given in UTF-8 encoding. For changing texts using plain ASCII-encoding, please refer to functions CmdSetCharIOx above.

**CmdQueueCharIO2 <new text>**  
**CmdQueueCharIO3 <new text>**  
**CmdQueueCharIO4 <new text>**  
**CmdQueueCharIO5 <new text>**

Enqueues a new ASCII text to be used for all input elements that make use of the logical "Char" inputs. When such an input element is assigned to a text or barcode primary element, the text or barcode is replaced by the given <new text> for the next marking operation. The four commands for char IO 2..5 correspond to the inputs 2..5 that can be selected in used input elements. Different to CmdSetCharIOx this command can be called several times right behind each other. On every call of this command, the new text is enqueued internally for automatic use on every marking cycle or on every increase of the current serial number. So with this command there is no need to implement a permanent handshake when text values have to be changed on every marking cycle, this will be done automatically with all the data buffered by each of these calls. When there are no more data left, the last text



is used permanently. When the serial number counter is reset or when a new project is loaded, all previously enqueued data are dropped.

The `<new text>` has to be given in plain 8 bit ASCII (local default encoding). For enqueueing texts in Unicode-encoding, please refer to functions `CmdSetUniIOx` below.

```
CmdQueueUniIO2 <new text>
CmdQueueUniIO3 <new text>
CmdQueueUniIO4 <new text>
CmdQueueUniIO5 <new text>
```

Enqueues a new UTF-8 text to be used for all input elements that make use of the logical "Char" inputs. When such an input element is assigned to a text or barcode primary element, the text or barcode is replaced by the given `<new text>` for the next marking operation. The four commands for char IO 2..5 correspond to the inputs 2..5 that can be selected in used input elements.

Different to `CmdSetUniIOx` this command can be called several times right behind each other. On every call of this command, the new text is enqueued internally for automatic use on every marking cycle or on every increase of the current serial number. So with this command there is no need to implement a permanent handshake when text values have to be changed on every marking cycle, this will be done automatically with all the data buffered by each of these calls. When there are no more data left, the last text is used permanently. When the serial number counter is reset or when a new project is loaded, all previously enqueued data are dropped.

The `<new text>` has to be given in UTF-8 encoding. For enqueueing texts in Unicode-encoding, please refer to functions `CmdSetCharIOx` above.

#### **CmdLaserOn <pen>**

This command turns on the laser for the currently selected head immediately and without dependence of any vector data. The parameter handed over together with this command specifies a pen which has to be used for controlling the laser. To use this command, the scanner controller device needs to be opened. This can be done with command `CmdOpenDevs`. To select a head, command `CmdSelHead` can be used.

#### **CmdLaserOff**

This command turns off the laser for the currently selected head immediately and without dependence of any vector data. When the laser has been turned on before by calling `CmdLaserOn`, it is mandatory to use this command before the device is closed with `CmdCloseDevs`. Elsewhere the state of the laser after closing the device is undefined. To select a head, command `CmdSelHead` can be used.

#### **CmdGetCardInpVal <inputflag>**

Read and return the status of a selected input of the card that belongs to the currently selected head. When this card does not have such an input or when an illegal value is given for parameter `inputflag`, an error is returned. The parameter itself can have exactly one of these values which specifies the input to be read:

- `IOCTRL_ANALOGUE_8_1` – 4
- `IOCTRL_ANALOGUE_8_2` – 8
- `IOCTRL_ANALOGUE_10_1` – 16
- `IOCTRL_ANALOGUE_10_2` – 32
- `IOCTRL_ANALOGUE_10_3` – 64
- `IOCTRL_ANALOGUE_10_4` – 128
- `IOCTRL_ANALOGUE_10_5` – 256
- `IOCTRL_ANALOGUE_10_6` – 512
- `IOCTRL_ANALOGUE_12_1` – 1024

- `IOCTRL_ANALOGUE_12_2` – 2048
- `IOCTRL_ANALOGUE_12_3` – 4096
- `IOCTRL_DIGITAL_8_1` – 8192
- `IOCTRL_DIGITAL_8_2` – 16384
- `IOCTRL_DIGITAL_16_1` – 32768
- `IOCTRL_DIGITAL_16_2` – 65536
- `IOCTRL_DIGITAL_32` – 131072
- `IOCTRL_ANALOGUE_16` – 262144
- `IOCTRL_ANALOGUE_12_4` – 524288
- `IOCTRL_ANALOGUE_16_1` – 1048576
- `IOCTRL_ANALOGUE_16_2` – 2097152

To select the head and therefore the card this command has to be used with, prior to calling this command `CmdSelHead` can be used.

#### **CmdSetCardOutpVal**

This command has to be used before `CmdSetCardOutp`, it is used to set a value which has to be sent to one of the outputs of a connected controller.

#### **CmdSetCardOutpMask**

This command can be used before `CmdSetCardOutp`, it is used to set a mask-value which has to be sent to one of the outputs of a connected controller.

#### **CmdSetCardOutp**

Using this command a value is sent to an output of the currently connected scanner controller card. This value has to be handed over by a preceding call to `CmdSetCardOutpVal` and mask is with an optional value set by a call to `CmdSetCardOutpMask`. It makes use of the card of the currently selected head (refer to `CmdSelHead`) and a single `IOCTL_`-flag which is handed over as parameter. When no mask is set or the mask is set to `0xFFFFFFFF`, then the full value is handed over to the controller. When a mask is set, only these bits of the value are changed at the output, which are specified by a “1” (=bit set) in the mask value.

The flag which has to be handed over specifies the output where the value has to be set to, here it depends on the currently used card which one is available and usable. The flags are the ones defined in header file `oapc_libio.h`. Following are supported (exclusively, it is not allowed to OR-concatenate them):

- `IOCTRL_LASERPORT_8_1` – 1
- `IOCTRL_LASERPORT_8_2` – 2
- `IOCTRL_ANALOGUE_8_1` – 4
- `IOCTRL_ANALOGUE_8_2` – 8
- `IOCTRL_ANALOGUE_10_1` – 16
- `IOCTRL_ANALOGUE_10_2` – 32
- `IOCTRL_ANALOGUE_10_3` – 64
- `IOCTRL_ANALOGUE_10_4` – 128
- `IOCTRL_ANALOGUE_10_5` – 256
- `IOCTRL_ANALOGUE_10_6` – 512

- IOCTRL\_ANALOGUE\_12\_1 – 1024
- IOCTRL\_ANALOGUE\_12\_2 – 2048
- IOCTRL\_ANALOGUE\_12\_3 – 4096
- IOCTRL\_DIGITAL\_8\_1 – 8192
- IOCTRL\_DIGITAL\_8\_2 – 16384
- IOCTRL\_DIGITAL\_16\_1 – 32768
- IOCTRL\_DIGITAL\_16\_2 – 65536
- IOCTRL\_DIGITAL\_32 – 131072
- IOCTRL\_ANALOGUE\_16 – 262144
- IOCTRL\_ANALOGUE\_12\_4 – 524288
- IOCTRL\_ANALOGUE\_16\_1 – 1048576
- IOCTRL\_ANALOGUE\_16\_2 – 2097152

When an invalid flag is handed over, the function call fails with error `OAPC_ERROR_INVALID_INPUT` immediately.

#### **CmdSendMark**

Using this command it is possible to send the current project data as stand-alone marking data to a connected scanner card. This requires a configuration where a scanner controller is set up properly, that supports this feature (like Scanlab RTC4 ScanAlone). Here no additional parameters are required since this function sends one exclusive set of marking data to the card.

#### **CmdWriteMark <path>**

Using this command it is possible to write the current project data as stand-alone marking data into a local file. This requires a configuration where a scanner controller is set up properly, that supports this feature (like HALaser E1701 or HALaser E1803 controller series). The given path specifies the full path of the file that has to be used for the stand-alone data. When the file already exists, it will be overwritten without further notice. Such a file can be transferred to the scanner controller manually in next step (e.g. by storing it onto the microSD card of E1701 and then putting this card back into the controller).

#### **CmdSendNamedMark <name>**

Using this command it is possible to send the current project data as stand-alone marking data to a connected scanner card using a given name. This requires a configuration where a scanner controller is set up properly, that supports this feature (like HALaser E1701 or HALaser E1803 controller series). The given <name> is an identifier the marking data are stored on the card with. Here the naming structure has to be according to the regulations of the used scanner card. For E1701 controller this means the name needs to be something like "0:/filename.epr" where "0:/" specifies microSD-card as storage place and ".epr" is the recommended file name extension for such stand-alone data. When a file with this name already exists on the controller, it will be overwritten without further notice.

#### **CmdSelEntName <name of element>**

Logically selects an element (but not highlights it in view). After an element with the given name was selected, it can be processed using other commands (as described below). In case the whole project was selected using `CmdSelPrj`, this selection is no longer valid.

#### **CmdSelEnt <id>**

Logically selects an element (but not highlights it in view). After an element with the given identifier was selected, it can be processed using other commands (as described below). In case the whole project was selected using `CmdSelPrj`, this selection is no longer valid.

#### **CmdSelPrj**

Logically selects the whole project including all elements (but does not highlight them in view). Afterwards this project and all of the contained elements can be processed using other commands (as described below). In case a single element was selected using `CmdSelEnt` or `CmdSelEntName`, this selection is no longer valid.

#### **CmdSetSelEnt <0/1>**

Select or deselect a selected element visibly in view. This operation is similar to a left-click on an element by the user while shift-key is held down. An additional parameter 0 or 1 decides if the element has to be selected or deselected. The element itself which is modified by this operation can be chosen by a call to function `CmdSelEntName` or `CmdSelEnt` which internally selects the element for further operations. After setting an element visibly selected by this command, all UI-operations can be performed that belong to selected elements. To fully zoom into a single element following sequence of commands can be used:

```
CmdSelEntName elementname
CmdSetSelEnt 1
CmdZoomElem
```

To zoom to two elements, this sequence of commands can be used:

```
CmdSelEntName elementname1
CmdSetSelEnt 1
CmdSelEntName elementname2
CmdSetSelEnt 1
CmdZoomElem
```

To unselect a previously selected element, parameter "0" has to be used

```
CmdSelEntName elementname1
CmdSetSelEnt 0
```

#### **CmdZoomElem**

Zooms view to fully show all selected elements. The element(s) to be shown after this command has/have to be selected by calling `CmdSelEntName/CmdSelEnt` and `CmdSetSelEnt` before.

#### **CmdDeleteEnt <id>**

Deletes an element completely. When the parameter id is set, the element with this identifier is deleted. When no parameter id is given, the currently selected element is deleted: This element has to be selected by calling `CmdSelEntName` or `CmdSelEnt` before. To make the element disappear from current view, a call to `CmdRedraw` may be necessary.

An example about how an element with the name "Rectangle 5" can be deleted:

```
CmdSelEntName Rectangle 5
CmdDeleteEnt
```

An example about how an element with the unique identifier 18 can be deleted:

```
CmdDeleteEnt 18
```

#### **CmdDeleteAllEnts**

Deletes all elements within the current project. Different to `CmdNewPrj` this command only removes the geometries but leaves hardware settings, pens and all possibly opened devices unchanged.

#### **CmdSetLoopRepeat <repeats>**

Changes the amount of repeats which have to be done for this element. A value of 0 means, the element is marked once and without repeating this operation. The element to be modified has to be selected by calling `CmdSelEntName` or `CmdSelEnt` before.

#### **CmdSetPosX <position>**

In case an element was selected by calling `CmdSelEntName` or `CmdSelEnt` before:

This command changes the X-position of the current element (left corner) to the value given as parameter in unit 1/1000 mm.

In case the whole project was selected by calling `CmdSelPrj` before:

This command translates the X-position of all elements of the current project by the given value in unit 1/1000 mm. Here "translation" means, the position value given with the command is added to the current position of all elements of the project, their positions are not overwritten by it.

To make the changes visible which have been applied by this command, a call to `CmdRedraw` may be necessary.

#### **CmdSetPosY <position>**

In case an element was selected by calling `CmdSelEntName` or `CmdSelEnt` before:

This command changes the Y-position of the current element (top corner) to the value given as parameter in unit 1/1000 mm.

In case the whole project was selected by calling `CmdSelPrj` before:

This command translates the Y-position of all elements of the current project by the given value in unit 1/1000 mm. Here "translation" means, the position value given with the command is added to the current position of all elements of the project, their positions are not overwritten by it.

To make the changes visible which have been applied by this command, a call to `CmdRedraw` may be necessary.

#### **CmdSetPosZ <position>**

In case an element was selected by calling `CmdSelEntName` or `CmdSelEnt` before:

This command changes the Z-position of the current element to the value given as parameter in unit 1/1000 mm.

In case the whole project was selected by calling `CmdSelPrj` before:

This command translates the Z-position of all elements of the current project by the given value in unit 1/1000 mm. Here "translation" means, the position value given with the command is added to the current position of all elements of the project, their positions are not overwritten by it.

To make the changes visible which have been applied by this command, a call to `CmdRedraw` may be necessary.

#### **CmdGetPosX**

Gets the X-position (horizontal) of the current element, the whole project or the output matrix and returns it using unit 1/1000 mm. The related element has to be selected by calling `CmdSelEntName` or `CmdSelEnt` or the project has to be selected by `CmdSelPrj` or the output matrix has to be selected by `CmdSelOutMtx` before.

#### **CmdGetPosY**

Gets the Y-position (vertical) of the current element, the whole project or the output matrix and returns it using unit 1/1000 mm. The related element has to be selected by calling `CmdSelEntName` or `CmdSelEnt` or the project has to be selected by `CmdSelPrj` or the output matrix has to be selected by `CmdSelOutMtx` before.

#### **CmdGetPosZ**

Gets the Z-position (depth) of the current element, the whole project or the output matrix and returns it using unit 1/1000 mm. The related element has to be selected by calling `CmdSelEntName` or `CmdSelEnt` or the project has to be selected by `CmdSelPrj` or the output matrix has to be selected by `CmdSelOutMtx` before.

#### **CmdSetScaleX <factor\*1000000>**

Changes the width of the current element by the factor given as parameter. The element to be modified has to be selected by calling `CmdSelEntName` or `CmdSelEnt` before. To make the changes visible which have been applied by this command, a call to `CmdRedraw` may be necessary.

#### **CmdSetScaleY <factor\*1000000>**

Changes the height of the current element by the factor given as parameter. The element to be modified has to be selected by calling `CmdSelEntName` or `CmdSelEnt` before. To make the changes visible which have been applied by this command, a call to `CmdRedraw` may be necessary.

#### **CmdSetScaleZ <factor\*1000000>**

Changes the depth of the current element by the factor given as parameter. The element to be modified has to be selected by calling `CmdSelEntName` or `CmdSelEnt` before.

#### **CmdGetScaleX**

Returns the scale factor which is applied to the width of the current element in unit \*1000. The element the scale factor has to be retrieved for needs to be selected by calling `CmdSelEntName` or `CmdSelEnt`.

#### **CmdGetScaleY**

Returns the scale factor which is applied to the height of the current element in unit \*1000. The element the scale factor has to be retrieved for needs to be selected by calling `CmdSelEntName` or `CmdSelEnt`.

#### **CmdGetScaleZ**

Returns the scale factor which is applied to the depth of the current element in unit \*1000. The element the scale factor has to be retrieved for needs to be selected by calling `CmdSelEntName` or `CmdSelEnt`.

#### **CmdGetSizeX**

Gets the X-size (width) of the current element and returns it using unit 1/10000 mm. The related element has to be selected by calling `CmdSelEntName` or `CmdSelEnt` before.

#### **CmdGetSizeY**

Gets the Y-size (height) of the current element and returns it using unit 1/10000 mm. The related element has to be selected by calling `CmdSelEntName` or `CmdSelEnt` before.

#### **CmdGetSizeZ**

Gets the Z-size (depth) of the current element and returns it using unit 1/10000 mm. The related element has to be selected by calling `CmdSelEntName` or `CmdSelEnt` before.

#### **CmdGetOutline**

Gets the complete outline (bounding box) of the current element and returns it using six values in unit 1/10000 mm. The returned values are separated by spaces and provide position X, position Y, position Z, size X, size Y and size Z consecutively in one single line. The element the outline has to be retrieved for needs to be selected by calling `CmdSelEntName` or `CmdSelEnt` before.

This is a convenience function which incorporates the functionality of commands `CmdGetPosX`, `CmdGetPosY`, `CmdGetPosZ`, `CmdGetSizeX`, `CmdGetSizeY` and `CmdGetSizeZ`.

#### **CmdSetRotX <angle>**

Changes the rotation around X-axis of the currently selected element/project to the value given as parameter in unit 1/1000 degrees. The element to be modified has to be selected by calling `CmdSelEntName` or `CmdSelEnt` before. When the whole project has to be rotated, it needs to be selected via `CmdSelPrj` before. To make the changes visible which have been applied by this command, a call to `CmdRedraw` may be necessary.

#### **CmdSetRotY <angle>**

Changes the rotation around Y-axis of the currently selected element/project to the value given as parameter in unit 1/1000 degrees. The element to be modified has to be selected by calling `CmdSelEntName` or `CmdSelEnt` before. When the whole project has to be rotated, it needs to be selected via `CmdSelPrj` before. To make the changes visible which have been applied by this command, a call to `CmdRedraw` may be necessary.

#### **CmdSetRotZ <angle>**

Changes the rotation around Z-axis of the currently selected element/project to the value given as parameter in unit 1/1000 degrees. The element to be modified has to be selected by calling `CmdSelEntName` or `CmdSelEnt` before. When the whole project has to be rotated, it needs to be selected via `CmdSelPrj` before. To make the changes visible which have been applied by this command, a call to `CmdRedraw` may be necessary.

#### **CmdGetRotX**

Gets the rotation around the X-axis of the current element, the whole project or the output matrix and returns it using unit 1/1000 degrees. The related element has to be selected by calling `CmdSelEntName` or `CmdSelEnt` or the project has to be selected by `CmdSelPrj` or the output matrix has to be selected by `CmdSelOutMtx` before.

#### **CmdGetRotY**

Gets the rotation around the Y-axis of the current element, the whole project or the output matrix and returns it using unit 1/1000 degrees. The related element has to be selected by calling `CmdSelEntName` or `CmdSelEnt` or the project has to be selected by `CmdSelPrj` or the output matrix has to be selected by `CmdSelOutMtx` before.

#### **CmdGetRotZ**

Gets the rotation around the Z-axis of the current element, the whole project or the output matrix and returns it using unit 1/1000 degrees. The related element has to be selected by calling `CmdSelEntName` or `CmdSelEnt` or the project has to be selected by `CmdSelPrj` or the output matrix has to be selected by `CmdSelOutMtx` before.

#### **CmdSetHatchAngle <angle>**

This function sets a hatch angle parameter in unit 1/1000 degrees. The value is set internally but not yet applied to any element. To do this, an additional call to `CmdSetHatch` is necessary.

#### **CmdSetHatchDist <distance>**

This function sets a hatch distance parameter in unit 1/1000 mm. The value is set internally but not yet applied to any element. To do this, an additional call to `CmdSetHatch` is necessary.

#### **CmdSetHatchOffs <offset>**

This function sets a hatch lines offset parameter in unit 1/1000 mm. The value is set internally but not yet applied to any element. To do this, an additional call to `CmdSetHatch` is necessary.

#### **CmdSetHatchPen <pen-number>**

This function sets a hatch pen. The value is set internally but not yet applied to any element. To do this, an additional call to `CmdSetHatch` is necessary.

#### **CmdSetHatchStyle <style-id>**

This function specifies the hatch style to be used. It can be one of the following text values:

- “fwd” – separate lines, forwarded
- “fwdbwd” – separate lines forward and backward
- “connected” – connected lines forward and backward
- “zigzag” – connected zigzag lines
- “inner” – inner lines repeating the shape of the element-specific

The value is set internally but not yet applied to any element. To do this, an additional call to `CmdSetHatch` is necessary.

#### **CmdSetHatchLinemode <mode-id>**

This function specifies the mode of the hatch lines to be used. It corresponds with the “linelength” parameter to be set with function `CmdSetHatchLinelen` and can be one of the following text values:

- “continuous” – hatch lines consist of single lines with one start and endpoint each, parameter “linelength” is ignored in this case
- “lines” – hatch lines consist of short lines combined to one longer, continuous line, parameter “linelength” specifies the length of the short line segments
- “dashed” – hatch lines are dashed lines, parameter “linelength” specifies the length of the lines and the distances between them
- “dotted” – hatch lines consist of single dots, parameter “linelength” specifies the distance between them
- “cust” – hatch lines consist of custom dots as specified in current projects settings, parameter “linelength” specifies the distance between the centre point of each of these custom dots



The value is set internally but not yet applied to any element. To do this, an additional call to `CmdSetHatch` is necessary.

#### **CmdSetHatchLinelen <length>**

This function sets the length of a hatch line which is used when a custom line mode is used (please refer to function `CmdSetHatchLinemode` above for details). The value is set internally but not yet applied to any element. To do this, an additional call to `CmdSetHatch` is necessary.

#### **CmdSetHatch <id>**

When this command is executed, the previously set hatch parameter are applied to an element. This element has to be selected by calling `CmdSelEntName` or `CmdSelEnt` before. To make the changes visible which have been applied by this command, a call to `CmdRedraw` may be necessary.

When no parameter `id` is given, a new hatch element is added to the selected element. The ID of the new element is returned. When the identifier of an existing hatch element is handed over as parameter `id`, this hatch element is modified and the new parameters are applied to it.

The returned value is the unique identifier for the hatch element and can be used with commands like `CmdDeleteEnt`, `CmdSelEnt` or `CmdSetHatch`.

An example about how to set a new hatch with 0,5 mm hatch distance, 45 degrees hatch angle and pen number 2 to an existing element "Circle 1":

```
CmdSetHatchDist 500
CmdSetHatchAngle 45000
CmdSetHatchPen 2
CmdSelEntName Circle 1
CmdSetHatch
```

An example about how to modify an existing hatch element with the identifier 8 to use 0,1 mm hatch distance, 180 degrees hatch angle and pen number 3:

```
CmdSetHatchDist 100
CmdSetHatchAngle 180000
CmdSetHatchPen 3
CmdSetHatch 8
```

Two examples about how to remove an existing hatch element with the identifier 8 from a parent element:

```
CmdSelEnt 8
CmdDeleteEnt
```

or

```
CmdDeleteEnt 8
```

#### **CmdRefresh**

Refresh the current view. This updates the shown geometries without recalculating and redrawing everything new

#### **CmdRedraw**

Redraw the current view. This performs a full re-calculation and re-drawing of all geometries and therefore is slower than `CmdRefresh`. This command should be called only when vector data of elements of the current project have been changed and the changes are not yet visible.

#### **CmdListName <num>**

This command expects a positive number `num` as parameter and can be used to list all available elements with their name. To get all elements, it has to be called repeatedly with increasing numbers (starting from 0). For every call with a different number the related element name will be given. When there are no more elements, an error is returned.

#### **CmdListUID <num>**

This command expects a positive number `num` as parameter and can be used to list the UIDs of all available elements. To get all elements UIDs, this command has to be called repeatedly with increasing numbers (starting from 0). For every call with a different number the related element UID will be returned. When there are no more elements, an error is returned.

#### **CmdSetElemFile <path>**

Changes the file that is assigned to an element and replaces the related data. This command can be used e.g. with Scanner Bitmap Elements to replace the current image.

The `path` to the new file has to be given as parameter and needs to point to an existing file in appropriate file format.

The element where the data are replaced by this command has to be selected by calling `CmdSelEntName` or `CmdSelEnt` before.

#### **CmdSetElemName <name>**

Changes the name of an entity which is shown in entity panel and which is used for identifying an element. Please note: The names generated by BeamConstruct always are unique. When changing the name, the calling application has to ensure the name is non-ambiguous. When there would be two or more elements with the same name within a project, it is no longer possible to select them via their names in a defined way.

The new `name` of the element has to be given as parameter. The element itself has to be selected by calling `CmdSelEntName` or `CmdSelEnt` before.

#### **CmdSetElemFlag <name>**

Sets a flag to an element that was selected by calling `CmdSelEntName` or `CmdSelEnt` before. The flag itself is specified by the given `name` and is activated by this function. The name of the flag and its functionality depends on the type of element that is selected. For an overview of available flags please refer to section “5.21.4.1 Element Flags” below.

#### **CmdResetElemFlag <name>**

Resets a flag of an element that was selected by calling `CmdSelEntName` or `CmdSelEnt` before. The flag itself is specified by the given `name` and is deactivated by this function.

The name of the flag and its functionality depends on the type of element that is selected. For an overview of available flags please refer to section “5.21.4.1 Element Flags” below.

#### **CmdSetElemText <text>**

Changes the text of a barcode or text primary element to the value given as parameter. The barcode or text element this new value is set for, has to be selected by calling `CmdSelEntName` or `CmdSelEnt` before. In case the GUI is visible, the element typically does not change its appearance immediately after this command is set. Although the internal data already have changed after this call, an additional call `CmdRefresh` is necessary to update the visual representation.

#### **CmdGetElemText**

Returns the current text of a barcode or text primary element. The barcode or text element the current text has to be retrieved for, has to be selected by calling `CmdSelEntName` or `CmdSelEnt` before. This function only returns the current text associated to this element but does not modify it in any way.

#### **CmdSetElemTextHeight <um>**

Sets a new nominal height for the currently selected text primary element. The new height has to be given as parameter in unit micrometers. The text element this new value is set for, has to be selected by calling `CmdSelEntName` or `CmdSelEnt` before. In case the GUI is visible, the element typically does not change its appearance immediately after this command is set. Although the internal data already have changed after this call, an additional call `CmdRefresh` is necessary to update the visual representation.

#### **CmdGetElemTextHeight**

Returns the current nominal height of the text within a text primary element. The text element the current text height has to be retrieved for, has to be selected by calling `CmdSelEntName` or `CmdSelEnt` before. The height value is returned in unit micrometers.

#### **CmdGetPen**

Returns the pen-number that is assigned to a selected element. The element where this command retrieves the pen-number from, has to be selected by calling `CmdSelEntName` or `CmdSelEnt` before.

#### **CmdGetPenFreq <pen-number>**

Returns the frequency in unit Hz that is assigned to the pen with the given number and to the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

#### **CmdGetPenPower <pen-number>**

Returns the power in unit 1/1000% that is assigned to the pen with the given number and to the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

#### **CmdGetPenColour <pen-number>**

Returns the RGB-colour that is assigned to the pen with the given number, to the head selected with command `CmdSelHead` (or head 0 by default) and which is used for drawing mark lines. Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**`CmdGetPenOffColour <pen-number>`**

Returns the RGB-colour that is assigned to the pen with the given number, to the head selected with command `CmdSelHead` (or head 0 by default) and which is used for drawing jump lines when the related option is enabled. Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**`CmdGetPenJSpeed <pen-number>`**

Returns the jump speed in unit  $\mu\text{m}/\text{sec}$  that is assigned to the pen with the given number and to the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**`CmdGetPenMSpeed <pen-number>`**

Returns the mark speed in unit  $\mu\text{m}/\text{sec}$  that is assigned to the pen with the given number and to the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**`CmdGetPenLOffDelay <pen-number>`**

Returns the laser-off-delay in unit  $\mu\text{sec}$  that is assigned to the pen with the given number and to the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**`CmdGetPenLOnDelay <pen-number>`**

Returns the laser-on-delay in unit  $\mu\text{sec}$  that is assigned to the pen with the given number and to the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**`CmdGetPenJDelay <pen-number>`**

Returns the jump-delay in unit  $\mu\text{sec}$  that is assigned to the pen with the given number and to the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**`CmdGetPenMDelay <pen-number>`**

Returns the mark-delay in unit  $\mu\text{sec}$  that is assigned to the pen with the given number and to the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**`CmdGetPenPDelay <pen-number>`**

Returns the in-polygon-delay in unit  $\mu\text{sec}$  that is assigned to the pen with the given number and to the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**`CmdGetPenSpotsize <pen-number>`**

Returns the lasers spotsize in unit  $1/10 \mu\text{m}$  that is assigned to the pen with the given number and to the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**`CmdGetPenWobbleAmp <pen-number>`**

Returns the wobble amplitude in unit  $\mu\text{m}$  that is assigned to the pen with the given number and to the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**`CmdGetPenWobbleFreq <pen-number>`**

Returns the wobble frequency in unit  $1/1000 \text{ Hz}$  that is assigned to the pen with the given number and to the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**`CmdGetPenWobbleAmpFuzz <pen-number>`**

Returns the wobble amplitude fuzzign value in unit  $\pm\mu\text{m}$  that is assigned to the pen with the given number and to the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**`CmdGetPenWobbleFreqFuzz <pen-number>`**

Returns the wobble frequency fuzzing value in unit  $\pm 1/1000 \text{ Hz}$  that is assigned to the pen with the given number and to the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**`CmdGetPenWobbleUpdateFreq <pen-number>`**

Returns the wobble update frequency in unit  $1/1000 \text{ Hz}$  that is assigned to the pen with the given number and to the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**`CmdGetPenWobbleMode <pen-number>`**

Returns the wobble mode that is assigned to the pen with the given number and to the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdGetPenPLength <pen-number>**

Returns the pulse length in unit  $\mu\text{sec}$  that is assigned to the pen with the given number and to the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdGetPenFPK <pen-number>**

Returns the first pulse killer pulse length in unit  $\text{nsec}$  that is assigned to the pen with the given number and to the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdGetPenStdByPLength <pen-number>**

Returns the stand-by pulse length in unit  $\mu\text{sec}$  that is assigned to the pen with the given number and to the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdGetPenValues <pen-number>**

Returns a list of all pen-parameters in the order of the single `CmdGetPenXXX`-commands listed above using the given pen number and the head which can be selected with command `CmdSelHead`. Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdSetPenFreq <pen-number> <frequency>**

Sets the *frequency* (in unit  $\text{Hz}$ ) to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdSetPenPower <pen-number> <power>**

Sets the *power* (in unit  $1/1000\%$ ) to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdSetPenColour <pen-number> <colour>**

Sets the *RGB-colour* which is used for drawing mark lines to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdSetPenOffColour <pen-number> <colour>**

Sets the *RGB-colour* which is used for drawing jump lines to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdSetPenJSpeed <pen-number> <speed>**

Sets the *jump speed* (in unit  $\mu\text{m/sec}$ ) to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdSetPenMSpeed <pen-number> <speed>**

Sets the mark *speed* (in unit  $\mu\text{m}/\text{sec}$ ) to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdSetPenLOffDelay <pen-number> <delay>**

Sets the laser-off-delay (in unit  $\mu\text{sec}$ ) to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdSetPenLOnDelay <pen-number> <delay>**

Sets the laser-on-delay (in unit  $\mu\text{sec}$ ) to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdSetPenJDelay <pen-number> <delay>**

Sets the jump-delay (in unit  $\mu\text{sec}$ ) to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdSetPenMDelay <pen-number> <delay>**

Sets the mark-delay (in unit  $\mu\text{sec}$ ) to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdSetPenPDelay <pen-number> <delay>**

Sets the in-polygon-delay (in unit  $\mu\text{sec}$ ) to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdSetPenSpotsize <pen-number> <spotsize>**

Sets the lasers *spotsize* (in unit  $1/10 \mu\text{m}$ ) to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdSetPenWobbleAmp <pen-number> <amp>**

Sets the wobble amplitude (in unit  $\mu\text{m}$ ) to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdSetPenWobbleFreq <pen-number> <frequency>**

Sets the wobble frequency (in unit  $1/1000 \text{ Hz}$ ) to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdSetPenWobbleAmpFuzz <pen-number> <amp>**

Sets the wobble amplitude fuzzing value (in unit  $\pm\mu\text{m}$ ) to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`. The value specified here has to be smaller than the value specified via command `CmdSetPenWobbleAmp`.

**CmdSetPenWobbleFreqFuzz <pen-number> <frequency>**

Sets the wobble frequency fuzzing value (in unit  $\pm 1/1000\text{ Hz}$ ) to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`. The value specified here has to be smaller than the value specified via command `CmdSetPenWobbleFreq`.

**CmdSetPenWobbleUpdateFreq <pen-number> <frequency>**

Sets the wobble update frequency (in unit  $\pm 1/1000\text{ Hz}$ ) to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`. This value specifies how often the wobble parameters have to be changed (according to what is set via `CmdSetPenWobbleAmpFuzz` and `CmdSetPenWobbleFreqFuzz`) in case a non-static wobble mode is used.

**CmdSetPenWobbleMode <pen-number> <mode>**

Sets the wobble mode to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`. This value specifies the mode wobbling has to take place with:

- when set to 0 (and/or when all fuzz-parameters or the update frequency is set to 0), straight wobble with no variances in wobble calculation is done
- when set to 1 and when at least one one of the fuzz-parameters is set and when a wobble update frequency is specified, the wobble-parameters are changed randomly resulting in some fuzzy wobble

**CmdSetPenPLength <pen-number> <length>**

Sets the pulse length (in unit  $\mu\text{sec}$ ) to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdSetPenFPK <pen-number> <length>**

Sets the first pulse killer pulse (FPK) length (in unit  $\text{nsec}$ ) to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdSetPenStdByPLength <pen-number> <length>**

Sets the stand-by pulse length (in unit  $\mu\text{sec}$ ) to the pen with the given number and the head selected with command `CmdSelHead` (or head 0 by default). Which pen is assigned to a specific element can be evaluated by calling function `CmdGetPen`.

**CmdSelHead <head-number>**

Sets the number of a head which has to be used for a subsequent command. Here head-number is a 0-based value that specifies a special head in multihead-environments. When this function is never called, all commands that are head-dependent, use head number 0 by default.



**CmdSetWAX <position>**

Set the X-position of the current working area using unit  $\mu\text{m}$ . The head this new value has to be applied for needs to be specified by a preceding call of CmdSelHead.

**CmdSetWAY <position>**

Set the Y-position of the current working area using unit  $\mu\text{m}$ . The head this new value has to be applied for needs to be specified by a preceding call of CmdSelHead.

**CmdSetWAZ <position>**

Set the Z-position of the current working area using unit  $\mu\text{m}$ . The head this new value has to be applied for needs to be specified by a preceding call of CmdSelHead.

**CmdSetWAWidth <size>**

Set the width (=horizontal size) of the current working area using unit  $\mu\text{m}$ . The head this new value has to be applied for needs to be specified by a preceding call of CmdSelHead.

**CmdSetWAHeight <size>**

Set the height (=vertical size) of the current working area using unit  $\mu\text{m}$ . The head this new value has to be applied for needs to be specified by a preceding call of CmdSelHead.

**CmdSetWADepth <size>**

Set the depth (=Z direction) of the current working area using unit  $\mu\text{m}$ . The head this new value has to be applied for needs to be specified by a preceding call of CmdSelHead.

**CmdSetEAX <position>**

Set the X-position of the global editing area using unit  $\mu\text{m}$ .

**CmdSetEAY <position>**

Set the Y-position of the global editing area using unit  $\mu\text{m}$ .

**CmdSetEAZ <position>**

Set the Z-position of the global editing area using unit  $\mu\text{m}$ .

**CmdSetEAWidth <size>**

Set the width (=horizontal size) of the global editing area using unit  $\mu\text{m}$ .

**CmdSetEAHeight <size>**

Set the height (=vertical size) of the global editing area using unit  $\mu\text{m}$ .

**CmdSetEADepth <size>**

This function returns the number of elements which exist in current project. Here not only primary elements are counted but every group, every groups sub-element(s), every hatch and every input element is counted as an own, separate one

#### **CmdGetElemNum**

Set the depth (=Z direction) of the global editing area using unit  $\mu\text{m}$ .

#### **CmdSetPilotMode <mode>**

This command corresponds to `TriggerUI 3021` and `TriggerUI 3022` and can be used to set the mode the pilot laser is working with. Following values are allowed to be set for `mode`:

- `PrjOutline` – let the pilot laser show the whole projects outline
- `ElemOutline` – let the pilot laser show every elements outline
- `ElemShape` – let the pilot laser show every elements real shape (without hatching)

#### **CmdSetPilotLoop <cnt>**

This command corresponds to `TriggerUI 3021` and `TriggerUI 3021` and can be used to set the number of loops the pilot laser has to perform at once. Here for `cnt` every value greater than 0 can be set.

#### **CmdCaptureView <width> <height> <path>**

Captures the drawing area and saves it as image into a file. Here `width` and `height` specify the size of the image to be saved. When positive values for both, `width` and `height` are specified, the image is scaled to that size. When one of both values is -1, the image is scaled using the value given for the other one by keeping the aspect ratio for that side that is set to -1. When both values are -1, the view is saved in it's original resolution.

`path` is the full path where the image has to be saved at. The file extension used for the name of the file in this path decides what format has to be used for saving. Supported file formats are `.png`, `.bmp`, `.gif` and `.jpeg` (while JPEG is not recommended to be used for unscaled images).

Example:

```
CmdCaptureView -1 -1 F:/viewcapture.png
```

will save the view in PNG format using it's original size on drive F: using file name `viewcapture.png`.

### **5.21.4.1 Element Flags**

This section describes the flags that are available for the different element types that can be set or reset with commands like `CmdSetElemFlag` and `CmdResetElemFlag`.

#### **5.21.4.1.1 Scanner Bitmap Primary Element Flags**

`ditherFS`, `bw`, `greyscale` – these flags are special, they can only be set and not reset and only one of them can be set. Means when a flag `bw` is set, the two other flags `ditherFS` and `greyscale` are automatically reset. These flags decide the format of the image which can be either a Floyd-Steinberg-dithered black-and-white image, a plain black-and-white image or a greyscale image

`mark_bidir` – this flag has an influence on behaviour during marking and enables a mode where lines are marked from beginning to end and from end to beginning alternating. This saves some marking time since the jump back to the beginning of the next line is saved.

`mark_fromlastline` – this flag has an influence on behaviour during marking and lets the image be processed from bottom to top when it is set

`mark_wo_lineincr` – this flag has an influence on behaviour during marking and marks all lines of an image at the same y-position

`mark_inverted` – this flag has an influence on behaviour during marking and inverts the power values resulting in an image that is negative to the visual representation; depending on the used material this finally may result in a positive image

`aspect_fixed_x`, `aspect_fixed_y` – these flags have no counterpart in BeamConstruct's UI and can be set or reset via BeamServer only; they are used when an image of an existing Scanner Bitmap Element is replaced by command `CmdSetElemFile`. When none of these flags is set, the new image is scaled to the same size like the previous image which may result in a loss of the current aspect ratio. When one of both flags is set, the aspect ratio of the new image is kept while one of the two sides X or Y uses the same width or height like the original image. When both flags are set, behaviour is undefined

#### **5.21.4.1.2 Z-Shifter Primary Element Flags**

The Z-shifter is a special variant of a motion element which consists of one axis only, which itself is used for moving a Z-axis. All flags of this elements make use of the prefix "X" which in this case does not mark the movement direction but the first (and only) axis. So this naming always

`XRel` – when this flag is set, a relative movement operation will be performed, without this flag, the given position will be used as absolute movement position

`XEnable` – this flag always has to be set in order to activate the axis for usage

`XHome` – when this flag is set, a homing operation will be performed instead of a defined movement to a position

#### **5.21.4.1.3 Motion Primary Element Flags**

`XRel` – when this flag is set, a relative movement operation will be performed with the first axis, without this flag, the given position will be used as absolute movement position

`XEnable` – this flag always has to be set in order to activate the first axis for usage

`XHome` – when this flag is set, a homing operation will be performed at the first axis instead of a defined movement to a position

`XStop` – when this flag is set, the related motion element forces a stop on the first axis for an already running movement

`YRel` – when this flag is set, a relative movement operation will be performed with the second axis, without this flag, the given position will be used as absolute movement position

`YEnable` – this flag always has to be set in order to activate the second axis for usage

`YHome` – when this flag is set, a homing operation will be performed at the second axis instead of a defined movement to a position

`YStop` – when this flag is set, the related motion element forces a stop on the second axis for an already running movement

`ZRel` – when this flag is set, a relative movement operation will be performed with the third axis, without this flag, the given position will be used as absolute movement position

`ZEnable` – this flag always has to be set in order to activate the third axis for usage

`ZHome` – when this flag is set, a homing operation will be performed at the third axis instead of a defined movement to a position

`ZStop` – when this flag is set, the related motion element forces a stop on the third axis for an already running movement

`URel` – when this flag is set, a relative movement operation will be performed with the fourth axis, without this flag, the given position will be used as absolute movement position

`UEnable` – this flag always has to be set in order to activate the fourth axis for usage

`UHome` – when this flag is set, a homing operation will be performed at the fourth axis instead of a defined movement to a position

`UStop` – when this flag is set, the related motion element forces a stop on the fourth axis for an already running movement

#### 5.21.4.1.4 Spiral Primary Element Flags

`ccw` – inverts the rotation direction of the spiral

`flatfront` – instead of having a spiral that is located flat on XY-plane, when this flag is set, the spiral is attached to the XZ-plane; this flag can't be combined with `flatright`

`flatright` – instead of having a spiral that is located flat on XY-plane, when this flag is set, the spiral is attached to the YZ-plane; this flag can't be combined with `flatfront`

`inToOut` – by default a spiral starts at the outside and shrinks to the inner, when this flag is set, this is reversed and the spiral is created from the centre point and grows to the outer diameter

#### 5.21.4.1.5 Circle Primary Element Flags

`ccw` – inverts the rotation direction of the circle

`flatfront` – instead of having a circle that is located flat on XY-plane, when this flag is set, the circle is attached to the XZ-plane; this flag can't be combined with `flatright`

`flatright` – instead of having a circle that is located flat on XY-plane, when this flag is set, the circle is attached to the YZ-plane; this flag can't be combined with `flatfront`

`laysphere` – when this flag is set, instead of a 2D, flat circle a 3D sphere is created which consists of different circles on different Z-positions that all together form the shape of a sphere

#### 5.21.4.1.6 Custom Output Primary Element Flags

`pulse` – when this flag is set, the output defined with this element is not set or reset but pulsed

#### 5.21.4.1.7 Barcode Primary Element Flags

`square` – this flag applies to some barcode types only, instead having a rectangular shape, when it is set, a squared shape is formed

`merge` – this flag applies to some 2D barcodes only, when set, the cells are merged in order to save geometry data and to speed up marking of the barcode

`invert` – when this flag is set, the barcode is inverted by surrounding it with a rectangle which – together with a hatch applied to this element – causes an inverted marking result

#### 5.21.4.1.8 Serial Number Input Element Flags

`resetAtTime` – with this flag set, the serial number counting is reset at the time given in the elements parameters

`resetAtValue` – with this flag set, the serial number counting is reset at the value given in the elements parameters

#### 5.21.4.1.9 Curve Postprocessing Element Flags

`keepElem` – with this flag set, when the underlying, bent primary element contains separate geometric parts, they are not bent on vector level, but are kept

`keepOrient` – with this flag set, when the underlying, bent primary element contains separate geometric parts, they are not rotated according to the bow of the generated curve, but are kept in their original orientation

### 5.22 Smart Interface

The interface described here is read-only. It can be used to implement Smart Factory (Industry 4.0) applications and provides different status informations which give the possibility to react on events during marking/production.

When only this interface is used, it is possible to watch processing of one or more BeamConstruct-controlled machines without the need to have an eye on them all the time. As soon as something important happens which requires user interaction, a notification is sent via the Smart Interface. This notification can be received by a suitable device (e.g. a Smartphone-application) which then informs the user that something has to be done at a specific machine. Now the operator can check that machine.

In combination with the BeamServer remote control interface or when programming interface of libbeamconstruct is used out of an own application, this interface gives extended possibilities to automate production with BeamConstruct according to the concept of smart factories. Now not only detailed state information are available but it is possible to react on them automatically. So e.g. when marking came to an end the application can decide if it starts the next marking operation or if it is necessary to have some operator interaction in order to continue production.

For more details about the BeamConstruct programming interface, please refer to the information about BeamSDK at <https://halaser.eu/feature.php#BeamSDK>.

#### 5.22.1 Accessing the Smart Interface

The smart interface data are provided via an unencrypted, unauthenticated TCP/IP socket using port number 11355. BeamConstruct sends several UTF-8-encoded XML documents via this interface. These documents provide state information but the interface does not accept any data sent by the connected client. So independent from what (unauthorized) client connects here, it is not possible to cause any dangerous operation via this connection.

This interface accepts an unlimited number of incoming connections. But to avoid an unnecessary high load on a running instance of BeamConstruct, it is recommended to not to connect with too much clients to this interface. For security reasons it is highly recommended to not to make this interface available via the Internet but only within a protected, walled-off production network.

The data provided at the Smart Interface all come in XML format and all have one common basic structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<BCMsg Time="2017-08-17T15:17:31" ID="901B0E5A06A2">
...
</BCMsg>
```

The first line is mandatory and specifies the beginning of a new XML document according to the XML specification. The second line starts the root-node `BCMsg` of every Smart Interface message. This root-node can contain two optional attributes `Time` specifying the time the message was sent (ISO 8601 time format, local time) and `ID` uniquely identifying the Machine/software installation this message was sent from.

Dependent on the kind of message which is sent, it may be extended by some more sub-nodes and attributes (symbolised by "..." in the example above).

Following the messages are described which are supported by the Smart Interface. PLEASE NOTE: although these messages are described as separate XML structures, they all can be combined within one `BCMsg`-node:

### Alive Message

This message is sent latest 40 seconds after last transmission in order to signal connected clients the software is still alive. When no messages have been received by a client for a longer time (e.g. for more than 150 seconds), the client should close the connection and try to reconnect. This alive-message does not provide any further information and therefore comes with an empty `BCMsg`-node:

```
<?xml version="1.0" encoding="UTF-8"?>
<BCMsg/>
```

### Status Message

This message is sent whenever the operational state of the application changes. It comes with a sub-node `Data` which contains an attribute `State`:

```
<?xml version="1.0" encoding="UTF-8"?>
<BCMsg>
  <Data State="<state>"/>
</BCMsg>
```

Here `<state>` can have following values:

`idle` – software is started but marking dialogue is not open and no mark operation is running

`ready` – mark dialogue is open but no marking operation is active yet

`marking` – a marking process is active

`paused` – a sliced 3D project was halted by pressing the “Pause”-button in mark dialogue; when this project is continued later and no error happens, it switches back to state “`marking`”

`error` – an error occurred, no marking is possible at the moment; in case this state is used, an additional Attribute `Text` provides the error text related to the current error (using the language the software is running with)

### Project Message

This message specifies which project is loaded at the moment. It comes with a sub-node `Data` which contains an attribute `Project`:

```
<?xml version="1.0" encoding="UTF-8"?>
  <BCMsg>
    <Data Project="<pathname>" />
  </BCMsg>
```

Here `<pathname>` is the full path to the project file which just was loaded. When it is empty, the "New" function was used and the last project was flushed completely.

## Immediate Trace Messages

This message is submitted as soon as an element is about to be marked where the "Trace data" option is set:

```
<BCMsg Time="2020-02-07T09:11:19" ID="B0C0904A5F15">
  <Trace Name="<name>" ID="<identifier>" />
</BCMsg>
```

The body encapsulates exactly one nodes of type `<Trace>` that contains an attribute `Name` which keeps the name of the element in project which is about to be marked. For input elements an additional attribute `ID` is contained which holds the input data which have been processed.

Please note: the information provided here only inform about the fact the current element is about to be marked. Since this software can't check if e.g. the working piece really was in place and was properly aligned, it does not necessarily mean the marking result was as expected, so additional inspection may be necessary with the working piece in order to ensure the correct result. Beside of that at this point it is also not known if the marking cycle was interrupted, possibly before the element announced with this message was processed. So to get a summary about the successful completion of a marking operation, refer to the "Number-Of-Parts Message" below.

## Number-Of-Parts Message

This message returns information about the current part number which has been produced and corresponds to the parts-parameter in global settings. When a `<parts>`-value of 0 is sent, the parts-counter was reset by the operator. Additionally there can be some optional attributes `ID` which uniquely identifies the current working piece and `ProdID` which uniquely identifies the product which is currently process. These identifiers typically are set externally, e.g. when BeamConstruct is integrated into an automated production line where previous machines provide these data via a suitable communication interface (like a Hermes connection):

```
<?xml version="1.0" encoding="UTF-8"?>
  <BCMsg>
    <Data Parts="<parts>" ID="<id>" ProdID="<prodid>">
      <Trace Name="<name>" />
      <Trace Name="<name>" ID="<identifier>" />
    </Data>
  </BCMsg>
```

The `<Data>`-section may encapsulate one or more nodes of type `<Trace>` when the "Trace data" option is enabled for elements in the current project. Every `<Trace>` section contains an attribute `Name` which keeps the name of the element in project which has been processed successfully by the last marking operation. For input elements an additional attribute `ID` is contained which holds the input data which have been processed. So this functionality can be used e.g. for traceability purposes. Since the Number-of-Parts message is emitted only in case a marking cycle was performed successfully, did not fail due to an error and was not interrupted by a stop-operation, the data contained in this message can be used for further

automated processing of the produced working piece. Together with an input element that provides unique data, it can be identified uniquely.

These trace information are also submitted immediately with an own message as soon as the related element is about to be marked. For more information please refer to the "Immediate Trace Message" above. Please note: the information provided here only inform about the fact a marking cycle has been performed successfully. Since this software can't check if e.g. the working piece really was in place and was properly aligned, it does not necessarily mean the marking result was as expected, so additional inspection may be necessary with the working piece in order to ensure the correct result.

### Cycle-Time Message

This message returns information about the time the last processing cycle took to complete. Additionally there can be some optional attributes `ID` which uniquely identifies the current working piece and `ProdID` which uniquely identifies the product this cycle time belongs to. These identifiers typically are set externally, e.g. when BeamConstruct is integrated into an automated production line where previous machines provide these data via a suitable communication interface (like a Hermes connection):

```
<?xml version="1.0" encoding="UTF-8"?>
<BCMsg>
  <Data CTime="<msecs>" ID="<id>" ProdID="<prodid>">
  </Data>
</BCMsg>
```

### Number-Of-Slice Message

This message is sent when a sliced 3D SLS/SLM model is built. Here the slices are announced as they are submitted to the scanner controller card. So this does not necessarily mean the current slice specified by value `<slice>` is finished, but it was handed over to the scanner card for production. The second attribute `MaxSlices` provides information about the maximum number of slices this model consists of:

```
<?xml version="1.0" encoding="UTF-8"?>
<BCMsg>
  <Data MaxSlices="<maxslices>" CurrSlice="<slice>"/>
</BCMsg>
```

### Process data Message

This message is sent when a process control plug-in is used and when this plug-in sends measured process data. These process data are provided via this interface:

```
<?xml version="1.0" encoding="UTF-8"?>
<BCMsg>
  <Process Name="<name>" Unit="<unit>" Idx="<index>" Value="<value>"/>
</BCMsg>
```

The data are encapsulated in a `Process`-tag which can exist several times within an XML package and which contains the following attributes:

`Name` – the name of the measured `<value>`

`Unit` – the measurement unit the `<value>` is given with



**Idx** – an index value in range 0..3; a plug in can measure up to four different values, this index specifies which of these values is provided

**Value** – the measured `<value>` which is further specified by the other attributes

## Notification message

Using this message, generic information can be submitted to a client. Such a notification message contains some text which is useful to the user and which can have different priorities:

```
<?xml version="1.0" encoding="UTF-8"?>
<BCMsg>
  <Msg Src="<sender>" Text="<shorttext>" Prio="<prio_number>"
    PrioText="<prio_text>" Desc="<detailtext>"/>
</BCMsg>
```

Different to the preceding ones this message provides different attributes within a `Msg`-tag:

- `Src` specifies the sender of this notification message, here either a unique machine identifier is provided (in case it is available) or a generic description
- `Text` is a short variant of the notification text, it acts as some kind of headline to give an overview what the notification is about
- `Desc` is the complete, detailed notification message
- `Prio` is a number and `PrioText` is a text corresponding to the number, both specify the priority this notification has:
  - 1 = "Fatal error"
  - 2 = "Error"
  - 3 = "Warning"
  - 4 = "Info"
  - all values >4 = "Other"

## 5.23 MQTT Data Interface

The interface described here is one-way only. The application connects to a MQTT broker (as configured in "5.6 Machine Configuration") and sends state information to this broker which then can be used by other clients which are connected to the same MQTT data broker.

When only this interface is used, it is possible to watch processing of one or more BeamConstruct-controlled machines without the need to have an eye on them all the time. As soon as something important happens which requires user interaction, a notification is sent via the MQTT Data Interface. This notification can be analysed by a suitable client application which then informs the user that something has to be done at a specific machine. Now the operator can check that machine.

In combination with the BeamServer remote control interface or when programming interface of libbeamconstruct is used out of an own application, this interface gives extended possibilities to automate production with BeamConstruct according to the concept of smart factories. Now not only detailed state information are available but it is possible to react on them automatically. So e.g. when marking came to an end the application can decide if it starts the next marking operation or if it is necessary to have some operator interaction in order to continue production.

For more details about the BeamConstruct programming interface, please refer to the information about BeamSDK at <https://halaser.eu/feature.php#BeamSDK>.

For more details about the MQTT protocol, please refer to <https://mqtt.org>

### 5.23.1 MQTT Messages

BeamConstruct sends several MQTT-messages to a connected broker automatically when a related event occurs. These messages provide state information but the interface does not accept any data sent by the broker BeamConstruct has connected to. So independent to what (unauthorized) client the software may connect to, it is not possible to cause any dangerous operation via this connection.

The data provided via the MQTT data interface are structured according to the MQTT specification and contain information as described below. Here the field "Topic" always contains a part `<basetopic>`, this is the topic as configured in "5.6 Machine Configuration". Unless otherwise noted all data come as plain ASCII-texts.

#### Status Message

This message is sent whenever the operational state of the application changes:

Topic: `<basetopic>/state/`

Data: `IDLE` – software is started but marking dialogue is not open and no mark operation is running

`READY` – mark dialogue is open but no marking operation is active yet

`MARKING` – a marking process is active

`PAUSED` – a sliced 3D project was halted by pressing the "Pause"-button in mark dialogue; when this project is continued later and no error happens, it switches back to state "MARKING"

`ERROR` – an error occurred, no marking is possible at the moment; in case this state is used, an additional error-message is submitted

#### Error Message

This message is sent whenever the operational state of the application changes to or from "ERROR":

Topic: `<basetopic>/error/`

Data: an error text in case a previous status message signalled an error state or an empty string in case no error happened or in case a previous error was reset

#### Project Message

This message specifies which project is loaded at the moment:

Topic: `<basetopic>/project/`

Data: the full path to the project file which just was loaded. When it is empty, the "New" function was used and the last project was flushed completely.

#### Number-Of-Parts Message

This message returns information about the current part number which has been produced and corresponds to the parts-parameter in global settings. When a value of 0 is sent, the parts-counter was reset by the operator:

Topic: `<basetopic>/parts/`

Data: the number of parts (as human readable ASCII-text)

### **Cycle-Time Message**

This message returns information about the time (in unit msec) the last processing cycle took to complete:

Topic: <basetopic>/ctime/

Data: the number of parts (as human readable ASCII-text)

### **Number-Of-Slice Message**

This message is sent when a sliced 3D SLS/SLM model is built. Here the slices are announced as they are submitted to the scanner controller card. So this does not necessarily mean the current slice is finished, but it was handed over to the scanner card for production:

Topic: <basetopic>/slices/

Data: <curr\_slice>/<max\_slices> - the slice information (as human readable ASCII-text) where  
    <curr\_slice> is the slice currently processed/sent to the controller and where <max\_slices> is  
    the total number of slices to be processed in current project.

## 6 CorrCorrect

This section describes CorrCorrect, a tool to create and modify scanhead correction tables.

### 6.1 Overview

When a scanhead – a system with two mirrors that deflect a laser in X and Y direction – is used, the resulting movement on the working area can be distorted. Here a few major reasons for distortions can be found:

1. a pincushion distortion that is caused by the mirrors and the used optics
2. a 3D focal distortion that happens because the distance between last mirror and working area differs dependent on the position where the laser hits it – the distance is shorter in the middle directly below of the mirror and longer on the borders of the working area, resulting in a loss of focus for some of the positions
3. a spatial correction because the scanhead is not mounted in parallel to working area because it has to work inside of a concave working piece

To correct such distortions special correction tables are used. In most cases these correction tables are specific to and delivered together with a scanhead. They use a vendor-specific file format that has to be converted depending on the used software/the used scanner controller plug-in. In other cases no such correction file exist and has to be generated from scratch. As a third case for some very high accuracy applications existing correction files have to be modified so that they fit to the used hardware exactly.

All these three tasks can be done with CorrCorrect correction file tool that is described here. The resulting correction table afterwards can be saved in BeamConstruct HD (.bco) correction file format – a special format that provides much better accuracy than most other formats. Such a correction file can be used for some of the scanner controller plug-ins of HAL software package directly (like the ETH6608 plug-in).

### 6.2 Usage

To start with generation of a new correction file either the menu-item “Project” → “Calculate new” or menu-item “Project” → “Interpolate new” has to be selected. Each of them opens a separate dialogue which provides a different method of generating a basic correction table.

**Calculate new:** Here a dialogue is opened where parameters for focal and pincushion distortion can be set. Additionally it can be specified which of both corrections have to be applied to the correction table. So for plain 2D applications or hardware set-ups where no third axis or no possibility to change the focus exists, only pincushion distortion parameters can be set. For a real 3D set-up focal correction parameters can be applied too. When none of the two options is set, a neutral correction file is generated. For a detailed description about how this dialogue can be used, please refer section “6.9 CorrCorrect Cookbook” below.

**Interpolate new:** When it is not intended to calculate a new correction based on some optical parameters, this option can be used. Here a dialogue opens, where the working area size and a grid of coordinates can be entered. This grid represents X and Y positions spread over the working area regularly. The grid is already filled with nominal coordinates that represent the positions to be marked. Now a similar Grid has to be marked without the use of any correction and then the real, actual marking positions of these points have to be entered into the related input fields. When clicking the OK-button, a correction file is generated which bases on these actual coordinates and contains a field, that corrects these positions to reach the nominal ones.

Alternatively it is possible to load an already existing correction file via menu “Project” → “Open” or by using toolbar button “Load Project”.

Next this correction table can be modified by specifying one or more nominal and actual positions. To get the values required for this, the current correction file has to be used for marking. During the marking operation one or more single points have to be marked. These points should be located at positions where the accuracy has to be increased or at positions spread over the whole working area. After marking the positions it has to be measured where the points have been marked really. The resulting values have to be entered in single or multiple point correction dialogue that can be opened via menu Edit → Correct Single Point or menu Edit → Correct Multiple Points. After entering the nominal coordinate(s) – the coordinates of the points within BeamConstruct – and the actual coordinate(s) – the real coordinates as measured after marking – the current correction table is modified at the given position(s) in order to reflect the deviation(s) at this/these point(s). This operation has to be repeated with the same or varied point coordinates until the resulting correction table offers the required accuracy.

**Please note:** This method of refining an existing correction table is intended to be used only when a correction file exists that is quite accurate. It can not be used to generate a new correction from scratch as it performs only some kind of fine-tuning close to the specified points but not adjusts the whole field in total. Please also note: these correction operations influence the correction table only in X- and Y-direction. A possibly existing focal correction is damaged by these operations. Because of that for full 3D correction the focal correction has to be applied again after this: By selecting menu Edit → Recalc Focal Correction the related correction is re-calculated according to the changed positions of 2D correction coordinates.

Additionally the CorrCorrect tool provides the same “Geometry”-panel like it is known from BeamConstruct. So when the current correction table – that is shown in the middle of the application – is selected, here several operations can be performed. It can be scaled, mirrored, centred, rotated, the total position can be changed and other things more.

## 6.3 Correction Calculation Dialogue

When menu “Project” → “Calculate new” is selected, a dialogue opens, where focal and pincushion distortion parameters can be set. There exists an other dialogue that provides only focal correction parameters and that can be selected via menu Edit → Recalc Focal Correction. This dialogues parameters are a subset of the parameters of the correction definition dialogue and therefore are not described separately.

This dialogue provides following general parameters:

- Z-Distance – the shortest distance between last mirror and working area

The correction dialogue provides following parameters for pincushion distortion:

- Checkbox Pincushion Distortion – the 2D / pincushion distortion parameters are applied to the newly generated correction table only when this checkbox is set
- X-angle – the distortion strength/angle in X direction
- Y-angle – the distortion strength/angle in Y direction

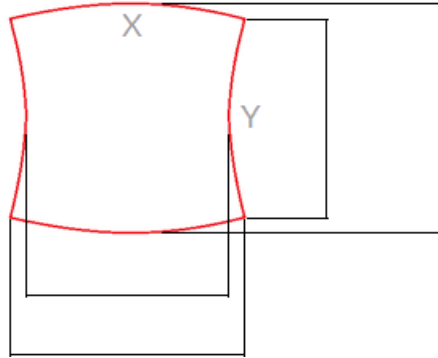
This dialogue provides following parameters for focal correction:

- Checkbox Focal Correction – the depth / focal distortion parameters are applied to the newly generated correction table only when this checkbox is set
- Working Area Size – the size of the working area; independent from its real shape here always a squared size is assumed and the biggest used size has to be entered here
- Maximum Z-Range – the maximum range the Z-axis is able to move; this value is typically smaller than the horizontal working range of the scanhead in X and Y direction and therefore has to be specified separately

### 6.3.1 Angle Calculation Wizard

As the X-angle and the Y-angle values out of the correction calculation dialogue (as described above) are not easy to evaluate, there is a wizard available which helps to get a first rough guess, what values are appropriate for an actual distortion. To access this wizard, first click one of the buttons “X-angle” or “Y-angle” (the text right beside the related input fields for the angle values).

Now an other dialogue opens, where distances have to be entered, which can be measured directly within the previously marked but still distorted square:



Here two values have to be entered for each of the directions X and Y: one for the distance of the corner points at the edges and one at the maximum/minimum expansion of the related edges. These values have to be entered into the related input fields. The measurement unit does not matter here, but needs to be exactly the same for all four fields.

After all four values have been entered, the wizard dialogue can be closed with “OK” and some suitable angles, calculated out of these measurements, are handed over to the parent correction calculation dialogue.

Please note: these angle values are quite good, but not yet perfect suitable. So they are a good starting point for generating a first correction, but they still need some fine-tuning afterwards (e.g. via function described in “6.8 Import Scanned Grid”).

## 6.4 Correction Interpolation Dialogue

When menu “Project” → “Interpolate new” is selected, a dialogue opens, where actual marking positions can be entered that are set in relation to expected nominal positions. The order to generate a correction file with this method is as follows:

1. Generate a BeamConstruct-project which marks dots, crosses or other appropriate geometries at the coordinates shown in the interpolation dialogue
2. Mark this project without any correction involved and measure the exact positions of these geometries
3. Enter the used working area size within the interpolation-dialogue
4. Enter all the measured, actual positions in the input fields that are preallocated with the nominal positions
5. Leave the Dialogue with “OK” to calculate a suitable correction

The entered coordinates are stored and available again when the application is started again in order to let the user continue working with already entered data. There are only two conditions under what the grid of actual coordinates is reset to its nominal defaults:

- the working area size is changed to a different value
- the button “Reset to nominal coordinates” is pressed

Please note: this method of generating a correction file is somewhat rough and therefore should be used only in cases where accuracy is not important or when additional steps are done in order to refine the resulting correction table. In all other cases it is highly recommended to use function “Calculate new” instead.

## 6.5 Correct Single and Multiple Points Dialogue

These two functions can be used to fine-tune an already existing correction table. So with them it is not possible to generate a complete correction from scratch but to do some fine-tuning on an existing correction file which is already close to be perfect. In other words: these functions are intended to be used in cases where an extreme accuracy is required which can't be provided by a correction file which is only calculated and does not base on real-world measurements at specific, critical points.

Both, the single and the multiple point dialogues expect the same values and behave the same except the following:

- the single point dialogue expects exactly one nominal and one actual coordinate value pair
- the multiple point dialogue asks for a CSV file that contains pairs of nominal and actual coordinate values and shows an editable list of them within the dialogue;  
The CSV-file contains comma-separated values in format  
`nominal x; nominal y; actual x; actual y;`  
using dot-separated floating point numbers

The dialogues offer the following fields and input values:

- Nominal Position X, Nominal Position Y – the nominal coordinate(s) in unit mm
- Actual Position X, Actual Position Y – the actual, measured coordinate(s) in unit mm
- Working Area Position X and Y – the top left position of the working area in unit mm
- Working Area Size – the size of the working area in unit mm, here a squared working area is assumed
- Influence scale – a factor which decides how much the correction at a single point has to influence the surrounding area in the correction table to be generated. Here a value of 1.0 means, the influence on surrounding area has to be as small as possible and will be exactly the difference between nominal and actual position. Every other value will be multiplied to the difference between these two points and expand the area around the nominal and actual point accordingly.  
When this scale factor is too small and the difference between nominal and actual position is too big, it may happen one correction grid point overlaps a previous/following correction grid point which would lead to undefined results and has to be avoided. In such a case the scale factor has to be increased.  
When a too big influence scale value is chosen in case of the multiple points dialogue, it may happen the influenced area is too big and overlaps a nearby nominal/actual position resulting in an over-compensation or invalid modification of this neighbour-point. In such a case the scale factor has to be decreased.

When the dialogue is left with "OK" the current correction table is modified according to the given nominal and actual value(s).

## 6.6 Spatial Correction Dialogue

This correction is intended for cases where the scanhead is not mounted in parallel to the working area but is inclined (e.g. in order to mark in inner side of a ring). It requires a correction that already fits to the used scanhead and optic, the spatial correction then can be applied to it as additional parameter. Thus following steps have to be performed in order to get an exact spatial correction:

- mount the scanhead in parallel to the working area in a way where the distance between scanhead and working area is the same length like the shortest distance between scanhead and working area when mounted inclined
- check if the correction for this scanhead fits and possibly modify it using the functions described above
- once you can mark an exact square that does not show any distortions, measure the size of that square and mount the head inclined

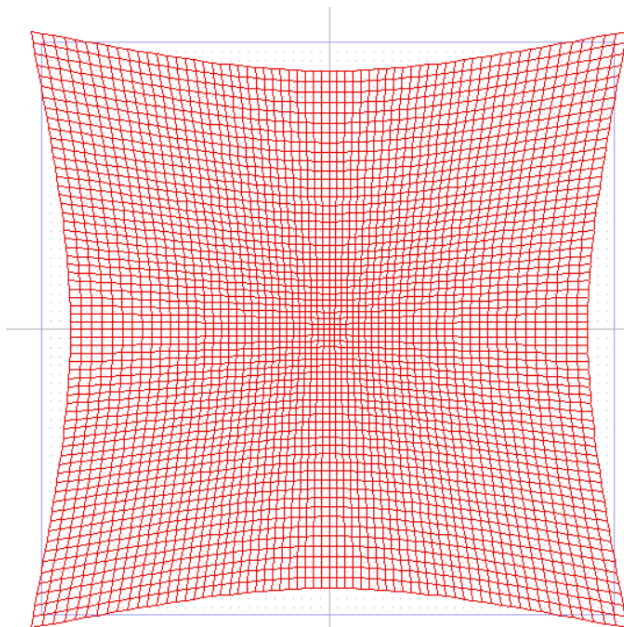
- mark the same square again; now that side of formerly square where distance between working area and scanhead is biggest has to be measured

The resulting two values – the normal (expected) length of the square measured in first step and the new, bigger length of the square measured after inclining the scanhead - have to be entered in dialogue for spatial correction. As an other information that side of the current correction table where the square has been distorted and where its edge is longer than before has to be marked using the radio buttons.

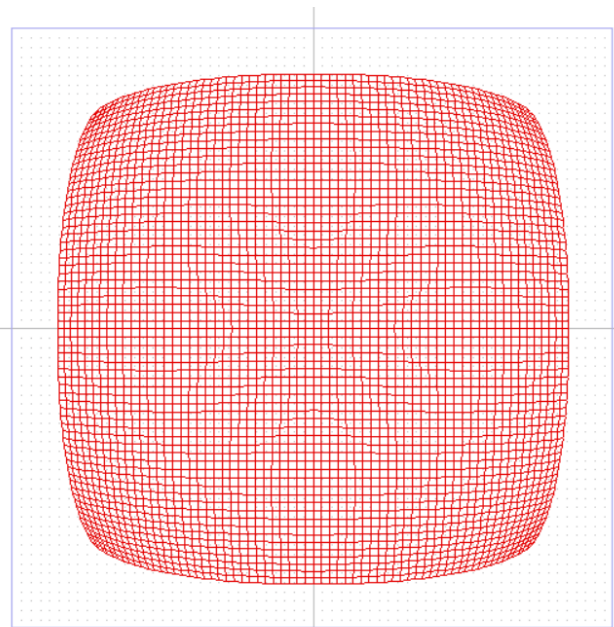
After pressing OK-button the spatial correction that stretched the squares width with growing distance will be applied to the correction table. Now the result is a rectangle. To correct its length to get back to a square, the gain factor of this direction has to be changed in scanner card settings (within BeamConstruct) or the width/height of the whole correction table has to be changed in Geometry-Panel of CorrCorrect directly.

## 6.7 Rising Scale Correction

After correcting the pincushion distortion it is possible, large geometries close to the outer borders of the working area itself are correct but the closer they are to the middle, they show an other distortion or their size is no longer exact. This can happen when the head shows a scaling effect which depends on the distance from the middle. Correction of this effect can be done via function in menu "Edit" → "Apply rising scale". It requires only one parameter which is a factor. As long as this factor is bigger than 1, it adjusts growing sizes depending on the distance to the centre point. A factor smaller than 1 adjusts sizes which become smaller dependent on the centre point distance. A factor of 1 would not apply any rising scale.



Rising scale correction applied to a neutral table using a scale factor  $>1,0$

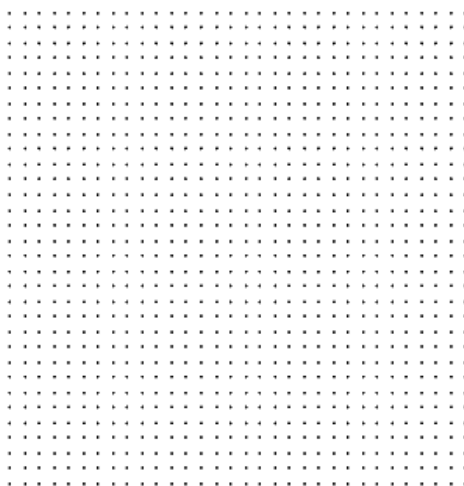


Rising scale correction applied to a neutral table using a scale factor  $<1,0$

## 6.8 Import Scanned Grid

CorrCorrect provides the possibility to update and enhance an existing base-correction by using an image of an array of dots that have been marked with this base correction which may look like this:





Correction enhancement based on such an image can be done with following simple steps:

- create a base-correction (as described in “6.9 CorrCorrect Cookbook”) or use the correction file provided by the vendor of your scanhead
- mark a regular, squared raster of dots with this correction (e.g. on a sheet of white paper or on a black surface), here all neighboured dots need to have exactly the same distance in X- and Y-direction, the grid needs to have the same number of dots in X- and Y-direction and there have to be dots in each of the four outer corner positions of the used working area (a suitable example can be downloaded at <https://halaser.systems/dotgrid.beamv>)
- digitize the marking result (e.g. by using a flatbed scanner) to get an image file
- in CorrCorrect load the base-correction this grid of dots was marked with
- go to menu “Edit” → “Apply scanned grid” and import the image from previous step; here all common raster image formats like PNG, BMP, GIF and others are supported

Now CorrCorrect loads this image, analyses the positions of the dots and adjusts the current correction table in order to reflect the deviations which still exist. After saving the modified correction table these steps can be repeated with that new correction to iteratively improve the quality of the result.

During analysis of the image, CorrCorrect uses the actual positions retrieved out of the imported image and the related nominal positions (which are defined by the total number of dots and the fact that they originally describe squares) to check the differences and to calculate what a correction table has to look like in order to equalise these deviations. Here CorrCorrect is able to adjust the typical pillow- or galvo-distortion as well as any other non-regular distortions. But it does not adjust offsets or a rotation of the scanhead along the Z-axis. These deviations later have to be adjusted either manually by adding them to the resulting correction table or by configuring the related offset- and rotation-parameters in settings of the used scanner controller card.

For the grid of dots following is important:

- the neighbouring dots need to form squares, so they all need to have the same distance in X- and Y-direction
- the upper left/upper right/lower left/lower right dot need to be located at the outer corners of the working area
- after marking, no vertical dots are allowed to overlap some other column’s vertical dots in X-direction and no horizontal dots are allowed to overlap some other row’s horizontal dots in Y-direction; when this is the case in resulting image, the base-correction needs to be improved or the distance between the dots needs to be increased
- there should be no dirt on the surface or scanned image which can be misinterpreted as dot

When some of these conditions are not met, the grid-detection-mechanism may not work properly and may misalign some of the scanned dots positions. This will result in a damaged correction table.

The overall accuracy of the resulting correction table can be influenced by following parameters:

- laser spot size: the laser spot size and resulting from that the marked dots should be as small and as sharply delineated as possible

- contrast: in both, the marked material and the scanned image there should be an as good as possible contrast, here it would be best to have black dots on a white surface or white dots on a black surface
- image quality: the scanned image should have an as high as possible resolution (large dpi-number) without any noise (noise can be misinterpreted as dots)
- rotation: the marked material should be aligned as exact as possible in both cases, when marked with the laser and when scanned in the flatbed scanner in order to not to have the dots containing a rotation which does not exist; the detection algorithm of CorrCorrect tries to automatically unrotate a grid of points, but this always results in additional inaccuracies due to rounding errors of the unrotation-function, so it is always a good idea to avoid such additional calculations in order to get a better accuracy.
- Image file format: do not use a file format which makes use of lossy compression algorithm (like in JPEG) as this changes the image, but something that has no or a lossless compression (like PNG, GIF or BMP)

When a scanned and loaded image is not accepted as grid and an error message appears, which tells, no squared grid was found, please check the following points:

- Does the used image contain transparency? In some images a transparency channel is used which – although it is transparent – has the same colour like the dots which makes it impossible to detect anything in the image. So please ensure there is no transparency in your image or make sure an image format is used which does not support transparency.
- Is there noise in the image, caused e.g. by some dirt on the marked sheet or by dust that appears on the scanned image? Such spots which are no regular marking dots can be misinterpreted, resulting in a grid with an illegal grid size. So please ensure a clean and accurate environment is used while creating such a scanned grid.
- What resolution is used? When scanning (aka digitising) the marked sheet, please use the best native resolution your flat-bed scanner offers in order to ensure the marked dots appear not only as single pixels in the image but as clear and large dots. This ensures they can be detected as wanted marking dots uniquely and are not filtered out as potential noise.

In case the grid detection failed, the related error message gives the possibility to show the detection result. When pressing “yes” to get this result, a dialogue opens where the original image (converted to greyscale) is shown. On top of that image the detected dots are drawn in green. Here one has to check:

- Are there some green markings in the image at positions where no dot was marked originally? This points to dirt on the surface or image which is misinterpreted as dot
- Are there dots in the original image which do not contain a green surrounding, means which have not been detected as regular dot? This points to the fact the resolution of the input image is too low so that the dots within the image are too small in order to be detected as suitable area.

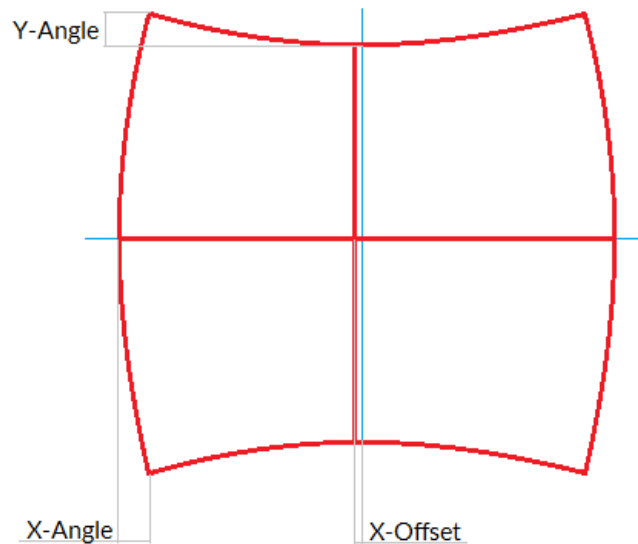
## 6.9 CorrCorrect Cookbook

Following some steps are described to get a suitable and accurate correction file that fits to a custom scanhead hardware set-up. Since every combination of scanhead with a specific lens results in an own, specific distortion, a separate correction file is needed for every scanhead-lens-combination. For the same combination typically the same correction file can be used.

By performing following steps a suitable correction file can be created:

1. Generate a project within BeamConstruct that consists of a square which is nearly as large as the whole working area (edge length should be about 96% of the maximum working area size that is possible with a scanhead and can be 100% in a second step when an already created correction file has to be verified and refined).
2. Add a horizontal and a vertical line to this project, both should be exactly centred vertically and horizontally to the working area, crossing each other in exact centre.

3. Mark this project on a material where the marking lines can be seen as clear and sharp as possible.
4. Check the marking result, the edges of the square and the position of the horizontal and vertical lines (please refer to example in image below).



Now switch over to CorrCorrect. When the square is distorted at top and bottom edge (distance marked in upper left corner in picture above), this distortion has to be corrected with the "Y-Angle" parameter in Correction Definition Dialogue. The stronger this deviation is, the larger the Y-Angle value should be.

When the square is distorted at left and right edge (distance marked in lower left corner in picture above), this distortion has to be corrected with the "X-Angle" parameter in Correction Definition Dialogue. The stronger this deviation is, the larger the X-Angle value should be.

To get some starting values for these two angles at least at the very first try, it is possible to use the angle calculation wizard as described in "6.3.1 Angle Calculation Wizard"

When the horizontal line is not marked at exact centre position of the working area (distance marked in lower middle position in picture above), after creation of a new correction file via Correction Definition Dialogue it has to be selected and shifted in vertical position. This can be done in tab-pane "Geometry" of CorrCorrect by moving it against the direction of its deviation (depending on the strength of distortion this line may also be bent a bit, in this case the distance from centre point has to be used for shifting).

When the vertical line is not marked at exact centre position of the working area, after creation of a new correction file via Correction Definition Dialogue it has to be selected and shifted in horizontal position. This can be done in tab-pane "Geometry" by moving it against the direction of its deviation (this deviation is not shown in example picture above).

In some cases it also may be necessary to rotate the correction table (e.g. when the marking result and its distortion is rotated by 90 degrees comparing to the example picture above). The rotation angle of 90 degrees can be entered in tab-pane "Geometry" once. Smaller/other rotation angles that compensate mounting accuracies should not be set here but by using parameter "Rotation" in scanner card settings.

5. Save the generated correction file, configure BeamConstruct to make use of it and mark the same geometry again. Depending on the result you may now continue with 4.) to refine the correction or finish here when the marked square does not show any distortions.

Geometry			
Mirror	X	Y	Z
Center	X	Y	Z
Scale	1.0000	1.0000	1.0000
Top Left	-32.2634	32.2634	0.0000
Bottom Right	32.2634	-32.2634	0.0000
Size	64.5268	64.5268	0.0000
	<input type="checkbox"/> Keep aspect ratio		
Rotation	-0.0000	-0.0000	-0.0000

Some important things that have to be mentioned for the process of creation of correction tables:

- There is no direct relation between the pincussion distortion of an edge of the marked square and the angle value to be set to adjust it, means there is no simple formula to calculate the angle-value out of a measured distance in marking result. Thus it is necessary to try in several steps to find out the correct values. When a given angle does not fully remove a distortion, it has to be set to a larger value in next step. When a used angle over-compensates a distortion and causes an edge to be bent in other direction, it has to be set to a smaller value.
- A correction table shown in CorrCorrect always has a base-size of 65,535 x 65,535 mm. This is an artificial size value that has to be used by CorrCorrect because this application does not know anything about used hardware set-up and real working area size. Thus a measured X- or Y-offset in real working area always has to be set in relation to this artificial working area size. The offset value to be entered within CorrCorrect then has to be a value calculated out of the real offset and the relation between real working area size in BeamConstruct and 65,535 mm artificial working area size in CorrCorrect.
- After saving a new correction file, it is mandatory to let BeamConstruct close a possibly existing connection to a scanner controller card. This can be done by going into project settings. BeamConstruct loads a correction file only once on initialisation of scanner controller card, thus a new initialisation has to be forced this way. When this is not done, BeamConstruct will continue working with a preceding correction table and will not notice there are any new data saved into the same file.

## 7 Firmware Downloader

The controllers of the E1701-series as well as the E1803D controller card all come with a firmware that it stored on a microSD-card. To update the firmware this card can be removed from the controller, placed into a PC using a suitable card adapter and then the update can be done simply by copying the files of the new firmware to the card.

In some cases this is not a suitable process when the microSD card or the controller are not easily accessible, e.g. because they are built into a machine. Here the firmware downloader FWDownloader can help: with this program it is possible to connect to a controller card, check the current firmware version and download a new one during the controller is running and without unmounting any hardware.

After starting the application, a window opens which provides the following options and functions:

- Controller – here the type of the connected controller has to be chosen. Available options for the used controller are “E1701A/C/D/M” for E1701A analogue scanner controller card, E1701C CNC controller, E1701D XY2-100 scanner controller card and E1701M stepper motor controller or “E1803D” for the E1803D XY2-100 scanner controller card.
- IP or serial interface – using this input field the communication interface of the selected controller can be chose. In case of an Ethernet connection here an IP in style aaa.bbb.ccc.ddd has to be entered, in case the controller is connected via USB, the assigned serial port (“COMx” or “/dev/ttyACMx”) has to be entered (where “x” is the number of the serial interface which is assigned to the controller)
- Ethernet password – when the controller is connected via Ethernet and when a password is configured in its e1701.cfg or e1803.cfg, this password needs to be entered here, elsewhere no communication with the card is possible
- Check current FW version – when this button is pressed, the FWDownloader connects to the controller and retrieves its current firmware version
- Get configuration – using this button the current configuration file e1701.cfg or e1803.cfg is uploaded and can be saved for further editing. Downloading of the modified file can be done with the following function:
- Download FW – here firmware files (consisting of e1701.fwi, e1701.dat, e1701.cfg and version.txt or consisting of e1803.fwi, e1803.dat, e1803.cfg and version.txt) can be downloaded to the controller. These files are stored on its microSD card and become active and used after next reboot. Please note: as downloading and writing of such files is a completely asynchronous process, after downloading a file, please wait about 5..10 seconds before rebooting the controller in order to ensure proper storage of the files.

## 8 BeamControl

As described above, BeamConstruct is a fully featured editor with loads of functions to create projects, manipulate geometries, modify parameters and even to control the whole process of marking. Although it contains a user management which gives the possibility to hide functions from users which are not privileged to do specific things, often it does not make sense to place this fully equipped software on a machine which does nothing more than marking already existing projects.

Here BeamControl comes into account (not to be confused with AndroidBeamControl which is an app for an Android-device and intended to watch running instances of BeamConstruct or even BeamControl). This software is stripped down to the minimum functionality that is needed for loading a project and marking it. It has a plain and clear user interface offering only these functions which are really needed, comes with big buttons and input fields that allow usage with a touch-screen and even provides a user management to not to let users start potential dangerous operations which are not privileged to do that.

BeamControl comes as separate executable which is located in HALsoftware installation directory (Windows) or in standard search path \$PATH (Linux). It is up to the machine integrator to start this application automatically and in foreground.

# Index

1	
11350.....	153, 155
11355.....	181
2	
2.5D.....	125
3	
3D.....	90f.
3D Manufacturing Format.....	125
3D Mesh.....	124, 131
3D printer format.....	125
3D Slice Cleaner.....	132
3D Studio.....	125
3D view.....	90
3DS.....	125
3MF.....	125
A	
About.....	17
AccuTrans 3D.....	125
activation threshold.....	83
Active Move.....	120
Active Move Group.....	122
Active Rotational Split.....	120
Active Split.....	120
Active Split Group.....	18, 120f.
Additional Geometry.....	87, 110
Additional laser control.....	23
Adjust projector.....	74
Alarm input (11).....	31, 36, 40, 44, 48
Alarm input (12).....	31, 36, 40, 44, 48
Alarm input (16).....	32, 36, 40, 44, 48
Alarm input (21).....	32, 36, 40, 44, 48
Align dots to raster.....	111
Align geometries at top or bottom.....	97
Align hatch to raster.....	112
Align left.....	120
Align right.....	120
Align spot to vectors.....	93
Align top.....	120
Alignment.....	93
Alive Message.....	182
All-over structures.....	130
Allow unrestricted laser on/off delays.....	24
Always on.....	19
analogue output.....	108
AndroidBeamControl.....	198
Angle Offset.....	112
Arrange in X and Y.....	97
Arrange in Z.....	97

Arrow to dot.....	94, 139
As project.....	76
aspect ratio.....	179
At border.....	113
Auto.....	139
Auto-arrange.....	96
Auto-Arrange.....	97
Auto-generate delays.....	77
Auto-move back to start position.....	22
Auto-optimize vectors on marking.....	24
Auto-slice 3D-meshes on change.....	24
axis alias.....	95, 139
Axis alias.....	22
Axis mapping.....	122f.

## B

Background Colour.....	18
Bar Width.....	105
Barcode.....	105
BCMMsg.....	181f.
bco.....	188
Beam Compensation.....	113
BeamConstruct.....	12
BeamConstruct HD.....	188
BeamConstruct to Control.....	107
BeamControl.....	198
Beamer.....	74
BEAMH.....	15f., 157
BeamLock.....	151
BEAMP.....	14f., 141, 157
BeamSDK.....	181, 185
BeamServer.....	153
BEAMV.....	15f., 157
Bezier Curve.....	102
Bind axis to slices Z-position.....	21
Bitmap greyscale mapping.....	85
Bitmap Pen Settings.....	85
Boolean add.....	124
Boolean intersect.....	124
Boolean subtract.....	124
Bottom To Top.....	126f.
Brightness.....	86
Broker.....	72

## C

Calculate new.....	188ff.
Calibrate Camera.....	133
calibration template.....	134
camera.....	134
camera calibration.....	134
Capture Image.....	62
Centre horizontal.....	120
Centre pt.....	91
Centre vertical.....	120
Centre X.....	91



Centre Y.....	91
Centre Z.....	91
chessboard.....	133
circle.....	25
Circle.....	99
Clean between slices.....	96, 126
Clean every x. slice.....	132
CLI.....	125
Close 3D mesh.....	124
Close lines.....	126f.
Close polygon with endpoint.....	124
CmdAppendPrj.....	159
CmdCaptureView.....	178
CmdCloseDevs.....	161
CmdClrElems.....	159
CmdDeleteAllEnts.....	164
CmdDeleteEnt.....	164
CmdGetCardInpVal.....	161
CmdGetElemNum.....	178
CmdGetElemText.....	171
CmdGetElemTextHeight.....	171
CmdGetEstMarkTime.....	160
CmdGetOutline.....	167
CmdGetPen.....	171
CmdGetPenColour.....	171
CmdGetPenFPK.....	174
CmdGetPenFreq.....	171
CmdGetPenJDelay.....	172
CmdGetPenJSpeed.....	172
CmdGetPenLOffDelay.....	172
CmdGetPenLOnDelay.....	172
CmdGetPenMDelay.....	172
CmdGetPenMSpeed.....	172
CmdGetPenOffColour.....	172
CmdGetPenPDelay.....	173
CmdGetPenPLength.....	174
CmdGetPenPower.....	171
CmdGetPenSpotsSize.....	173
CmdGetPenStdByPLength.....	174
CmdGetPenValues.....	174
CmdGetPenWobbleAmp.....	173
CmdGetPenWobbleAmpFuzz.....	173
CmdGetPenWobbleFreq.....	173
CmdGetPenWobbleFreqFuzz.....	173
CmdGetPenWobbleMode.....	173
CmdGetPenWobbleUpdateFreq.....	173
CmdGetPosX.....	165, 167
CmdGetPosY.....	166f.
CmdGetPosZ.....	166f.
CmdGetPrjName.....	159
CmdGetPrjPath.....	159
CmdGetRotX.....	167
CmdGetRotY.....	167
CmdGetRotZ.....	167

CmdGetScaleX.....	166
CmdGetScaleY.....	166
CmdGetScaleZ.....	166
CmdGetSizeX.....	166f.
CmdGetSizeY.....	166f.
CmdGetSizeZ.....	167
CmdImport.....	160
CmdImportSilent.....	160
CmdIsMarking.....	156
CmdLaserOff.....	161
CmdLaserOn.....	161
CmdListName.....	170
CmdListUID.....	170
CmdLoadPenPrj.....	159
CmdLoadPrj.....	159
CmdNewPrj.....	159, 165
CmdOpenDevs.....	161
CmdQueueCharIO2.....	160
CmdQueueCharIO3.....	160
CmdQueueCharIO4.....	160
CmdQueueCharIO5.....	160
CmdQueueUnilO2.....	161
CmdQueueUnilO3.....	161
CmdQueueUnilO4.....	161
CmdQueueUnilO5.....	161
CmdRedraw.....	169
CmdRefresh.....	169
CmdResetElemFlag.....	170, 178
CmdSavePenPrj.....	159
CmdSavePrj.....	159
CmdSelEnt.....	163
CmdSelEntName.....	163
CmdSelHead.....	161, 176
CmdSelPrj.....	164
CmdSendMark.....	163
CmdSendNamedMark.....	163
CmdSetCardOutp.....	162
CmdSetCardOutpMask.....	162
CmdSetCardOutpVal.....	162
CmdSetCharIO2.....	160
CmdSetCharIO3.....	160
CmdSetCharIO4.....	160
CmdSetCharIO5.....	160
CmdSetEADepth.....	177
CmdSetEAHeight.....	177
CmdSetEAWidth.....	177
CmdSetEAX.....	177
CmdSetEAY.....	177
CmdSetEAZ.....	177
CmdSetElemFile.....	170, 179
CmdSetElemFlag.....	170, 178
CmdSetElemName.....	170
CmdSetElemText.....	171
CmdSetElemTextHeight.....	171

CmdSetHatch.....	169
CmdSetHatchAngle.....	168
CmdSetHatchDist.....	168
CmdSetHatchLinelen.....	169
CmdSetHatchLinemode.....	168
CmdSetHatchOffs.....	168
CmdSetHatchPen.....	168
CmdSetHatchStyle.....	168
CmdSetLoopRepeat.....	165
CmdSetPenColour.....	174
CmdSetPenFPK.....	176
CmdSetPenFreq.....	174
CmdSetPenJDelay.....	175
CmdSetPenJSpeed.....	174
CmdSetPenLOffDelay.....	175
CmdSetPenLOnDelay.....	175
CmdSetPenMDelay.....	175
CmdSetPenMSpeed.....	175
CmdSetPenOffColour.....	174
CmdSetPenPDelay.....	175
CmdSetPenPLength.....	176
CmdSetPenPower.....	174
CmdSetPenSpotsize.....	175
CmdSetPenStdByPLength.....	176
CmdSetPenWobbleAmp.....	175f.
CmdSetPenWobbleAmpFuzz.....	176
CmdSetPenWobbleFreq.....	175f.
CmdSetPenWobbleFreqFuzz.....	176
CmdSetPenWobbleMode.....	176
CmdSetPenWobbleUpdateFreq.....	176
CmdSetPilotLoop.....	158, 178
CmdSetPilotLoop.....	158
CmdSetPilotMode.....	158, 178
CmdSetPosX.....	165
CmdSetPosY.....	165
CmdSetPosZ.....	165
CmdSetRotX.....	167
CmdSetRotY.....	167
CmdSetRotZ.....	167
CmdSetScaleX.....	166
CmdSetScaleY.....	166
CmdSetScaleZ.....	166
CmdSetSelEnt.....	164
CmdSetUnilO2.....	160
CmdSetUnilO3.....	160
CmdSetUnilO4.....	160
CmdSetUnilO5.....	160
CmdSetWADepth.....	177
CmdSetWAHeight.....	177
CmdSetWAWidth.....	177
CmdSetWAX.....	177
CmdSetWAY.....	177
CmdSetWAZ.....	177
CmdStopMark.....	156, 158

CmdWriteMark.....	163
CmdZoomElem.....	164
CNC.....	26, 28
Coherent(R) Avia Ethernet Laser Controller.....	63
Collapse all.....	96
Colour after marking.....	76, 86
Coloured Bitmap Marking.....	86
coloured scanner-bitmaps.....	76, 86
Combine with laser state.....	19
comma separated values.....	116
Common Layer Interface.....	125
Concatenated slices.....	132
Conditional offset.....	117
Contrast.....	86
Control Marking.....	73
Conveyor axis.....	72
Corner radius.....	99
CorrCorrect.....	188
Correct Multiple Points.....	189
Correct Single Point.....	189
Correction Blue.....	86
correction factor.....	19
Correction Factor.....	19
Correction Green.....	86
correction offset.....	19
Correction Offset.....	19
Correction Red.....	86
correction table.....	188
Count X/Y/Z.....	93
counter-clockwise.....	116
Create 3D mesh.....	124
CreateLog.....	156
Credits.....	17
Crop Camera.....	134
CSV.....	116
CSV Data Input.....	109f., 116
Curve Distortion.....	115
Custom dot.....	25
Custom dot name.....	25
Custom Dot Style.....	25
Custom Input.....	108
Custom Output.....	107
Cycle-Time Message.....	184, 187
<b>D</b>	
Data Input.....	116
Decrement.....	139
Delay.....	106
Delay after MO.....	32, 37, 41, 44, 49
Delay before MO.....	32, 36, 41, 44, 49
Delay before MO off.....	32, 37, 41, 44, 49
Delete Element(s).....	96
Derive cleaning shape from.....	132
Destructive.....	119

digital output.....	108
digital projector.....	74
Disable 3D view.....	71
Disable centre point.....	71
Display image on manual activation only.....	22
Display message during wait.....	108
Display reduced.....	128
Display Reduced.....	129
Distance to Model.....	130
Distance X/Y/Z.....	93
distorted image.....	134
Don't show warning when referencing axes.....	22
Don't wait for end of motion.....	107
Dot.....	97
Dot-length.....	86
Double arrows.....	94, 139
Down to bottom of editing area.....	131
Draw CCW.....	25, 100ff.
Draw Cells as Custom Dots.....	106
Draw Cells as Dots.....	106
Draw Off-Movements.....	75
Drop Calibration.....	134
Drop Fiducial.....	134
Drop home positions.....	19
Dummy scanner controller.....	25
Duplicate.....	96, 120
 E	
E1701A.....	29, 197
E1701C.....	33, 197
E1701D.....	37, 197
E1701M.....	56ff., 65ff., 197
E1702D.....	42
E1803D.....	45, 197
E1803D PID control loop.....	67
Edges/Segments.....	25
Edit Geometry.....	73
Editing area.....	18
Element Tab-Pane.....	92
Element Tree.....	115
ElemOutline.....	178
ElemShape.....	178
Enable Hermes automatic control.....	72
Enable Hermes remote configuration.....	73
Enable MQTT connection.....	72
Enable overspeed.....	84
Enable Smart Interface.....	72
Enable user management.....	73
Enable window dragging border.....	74
Enclose model completely.....	130
End Angle.....	100
Enlarge by.....	132
Enlarge Structure.....	130
Error Message.....	186

ETH6608.....	55, 188
Exit Application.....	73
ExitUI.....	153, 156
Expand all.....	96
Expand By Dot.....	98
Export as.....	16
Extend displays.....	74
External Trigger.....	109
Extrude.....	124
Extrusion axis.....	124
Extrusion direction.....	124
Extrusion length.....	124

## F

Fade In Size.....	105
Fade out references.....	93
fade-in distance.....	83
fade-out distance.....	84
False Colours.....	133
Fast repeat.....	138
fiducials.....	133
First slice.....	132
First Slice.....	128
Fixed height for jumps.....	28, 35
focal correction.....	188f.
Focal Correction.....	189
Follow-up time.....	20
font type.....	104
Font Type.....	103
Force to centre coordinate.....	116
Force to Square.....	105
Freeze Video.....	133
Frequency.....	75
From bounding box.....	113
From pen settings.....	93
front view.....	90
Fuzzy.....	78
FWDDownloader.....	197

## G

G-Code.....	26ff., 49
G-Code Controller (Ethernet).....	26, 28
G-Code Controller (Serial).....	28
G00.....	27
Gamma.....	86
General Pen Settings.....	75
Generate bugreport.....	17
Generic analogue temperature.....	68
Generic circle.....	132
Generic Laser Controller.....	63
Generic rectangle.....	132
Geometry.....	139
Geometry Tab-Pane.....	91
Go back from this result.....	76
GRBL.....	28

greyscale.....	85
Group elements.....	95
Grow.....	100, 102
<b>H</b>	
HALaser E1701A Scanner Controller:.....	29
HALaser E1701C CNC Controller:.....	33
HALaser E1701D Scanner Controller:.....	37
HALaser E1701M Focus Shifter Motor Controller.....	65
HALaser E1701M Stepper Motor Controller.....	56
HALaser E1702S Scanner Controller:.....	41
HALaser E1803D Scanner Controller:.....	45
HALaser Systems E1803D Motion (Intelli-IO controlled).....	58
Halt with ExtStart.....	40, 44, 48
Hatch Angle.....	112
Hatch Distance.....	112
Hatch Offset.....	112
Hatcher.....	111
Head.....	92
Hermes.....	72, 183f.
Hide hatch from view.....	115, 128
Hide support structure settings.....	71
home position.....	19
Home position.....	19
HPGL to Control.....	141
<b>I</b>	
Image capture.....	22
Immediate Trace Messages.....	183
immersive production.....	71
Import.....	15, 86
Import 3D model.....	96
IMS.....	59
In-Polygon Delay.....	78
Increment.....	139
Industry 4.0.....	72, 181
Initialise on startup.....	18
Inner Radius.....	25
Input Element.....	87
Input Number.....	20
Input Port.....	20
Insert project.....	15
Inside To Outside.....	25, 102
Intelli-IO.....	58
Interleave Lines.....	112
Interlock Input.....	20
Interpolate new.....	188, 190
Invert.....	106, 113
Invert Output Logic.....	19f.
Invert with embedded normal.....	106
IO Sync Motor Trigger (Scannercard).....	60
IOCTRL_LASERPORT_8_1 – 1.....	162
IOSelect Mode.....	143
IOSelect-Mode.....	23
IP Camera.....	62

Isel.....	5
<b>J</b>	
Jump.....	138
Jump Delay.....	76, 78
Jump Speed.....	75
Jump Speed.....	24
jump speed factor.....	84
Jump speed factor.....	84
<b>K</b>	
Kerning.....	104
Key controlled.....	22
Keystone correction.....	121, 123
<b>L</b>	
Language.....	71
Laser Off Delay.....	78, 85
Laser On Delay.....	78, 85
LaserB.....	107
Laserport.....	107
Last Slice.....	128
layer.....	94
Layer.....	93
Layer Tab-Pane.....	94
Layered Sphere.....	100
Lazy input switch.....	61
Letter set.....	119
Limits.....	24
Line.....	98
Load default configuration.....	16, 25
Load Fiducial.....	134
Load hardware settings.....	15
Load payload data.....	15
Load Project.....	73
load recipe.....	73
Lock size.....	105
lock-rules.....	151
<b>M</b>	
M7xx.....	27
Machine name.....	72
main oscillator.....	31, 36, 40, 43, 48
Main oscillator.....	108
Make use of Z-coordinate.....	126f.
Makeblock XY-Plotter.....	49
Manager Users.....	73
Manual.....	138
Mark.....	136
Mark contour first.....	93
Mark Delay.....	77f.
mark dialogue.....	137
Mark dialogue.....	64
Mark dialogue opened signal.....	20
Mark repeatedly.....	138, 144



Mark selected elements.....	138
Mark Speed.....	75
Mark Speed.....	24
Marking.....	137
Marking Active Signal.....	20
Maximum Z-Range.....	189
Menu item shortcuts.....	71
Merge.....	123
Merge 3D meshes.....	96
Merge Cells.....	106
Milling.....	28, 35
Mirror X.....	91
Mirror Y.....	91
Mirror Z.....	91
Misc.....	23
MO.....	31, 36, 40, 43, 48
Mode.....	138
Modify live background.....	22
Modify Settings.....	73
monospaced.....	105
MOTF-Distance.....	109
motion.....	110
Motion.....	94, 106, 139
Motion Axes.....	21
Motion Controller.....	21
Motion Tab-Pane.....	94
Move axes to home-position.....	107
Move down.....	96
Move entity tree to tab panel.....	71
Move to bottom.....	96
Move to top.....	96
Move up.....	96
MQTT.....	72, 185f.
MQTT Data Interface.....	185
multi-line text.....	103
multihead.....	18
Multihead.....	21, 92, 141
 N	
New project.....	15
Next machine.....	72
No license progress bar.....	71
No warning message at start-up.....	71
Nominal height.....	105
Non-Destructive.....	119
Notification message.....	185
Number of Lines.....	112
Number of vectors.....	93
Number-Of-Parts Message.....	183, 186
Number-Of-Slice Message.....	184, 187
 O	
OAPC_ERROR_CREATE_FILE_FAILED.....	155
OAPC_ERROR_DEVICE.....	155
OAPC_ERROR_INTERNAL.....	155

OAPC_ERROR_INVALID_INPUT.....	155
OAPC_ERROR_NO_MEMORY.....	155
OAPC_ERROR_OPEN_FILE_FAILED.....	155
OAPC_ERROR_READ_FILE_FAILED.....	155
OAPC_ERROR_RESOURCE.....	155
OAPC_ERROR_STILL_IN_PROGRESS.....	155
OAPC_ERROR_WRITE_FILE_FAILED.....	155
OAPC_NO_DATA_AVAILABLE.....	155
OBJ.....	125
Object File Format.....	125
OFF.....	125
Offset X.....	74
Offset Y.....	74
Open project.....	15
OpenAPC Data Input.....	117
OpenGL.....	71
Optimize.....	124
Optris LT Pyrometer.....	69
OUT2.....	21
OUT3.....	21
Outer lines rounded.....	112
Outer lines straight.....	112
Outer Radius.....	25
Output Number.....	19f.
Output Port.....	19f.
Overlap splits.....	121
Override manual stop.....	19
Overspeed.....	84
 <b>P</b>	
Parameter Wizard.....	76
Parts.....	139
Parts Counter.....	24
Pen.....	92
Pen Frequency.....	24
Pen Off-Colour.....	75
Pen On-Colour.....	75
Pen Parameter Wizard.....	76
pen settings.....	25
Pen settings.....	16
Pen Settings.....	74
pilot.....	138
pilot laser.....	73
Pilot laser.....	107
Pilot Laser.....	19, 138
pincushion distortion.....	188f.
Pincushion Distortion.....	189
PLY.....	125
point editing.....	89, 92
polygon.....	25
Polygon.....	102
Polygon Delay.....	77
Polygon Style.....	102
Position tolerance.....	28

Post-processing Tool.....	87
Power.....	75
Power down idle timeout.....	25
Power down laser.....	25
power range.....	85
Preferred head usage.....	21
Previous machine.....	72
Primary Geometry.....	87, 97
Printer driver controlled.....	50
PrjOutline.....	178
Process.....	131
Process between slices.....	21f.
Process Controller.....	23
Process data Message.....	184
Process element.....	93
Process order.....	121, 123
Process time.....	94
Program Flow.....	109
Progress.....	140
Project Message.....	182, 186
Project onto ground.....	91, 96
Project settings.....	16
projector.....	73
Pulse Output.....	108
PWM control.....	32, 37, 41, 45, 49

## R

Raylase(R) SP-ICE2.....	50
Ready For Marking Signal.....	19
Real 3D.....	126f.
Recalc Focal Correction.....	189
Rectangle.....	99
Reduce Geometry.....	124
Reduction.....	125, 127
Ref.....	95, 139
Reference axis on start-up.....	22
Reference copies.....	93
Reference Speed.....	59, 61
Referencing order.....	22
Refine this result.....	76
Reload on file change.....	16
Remote Control Commands.....	155
Remove Last Dot.....	98
Remove points in angles.....	124
RenderWare.....	125
Repeat marks.....	93
Require trigger.....	21
Reset.....	139f.
Reset controller after download.....	137
Reset Counter.....	139
Reset license.....	17
Resort elements.....	95
Resort Elements.....	96
Reverse direction.....	120

right view.....	90
Rotate CCW.....	116
Rotation.....	74, 92
RWX.....	125
<b>S</b>	
Save as default.....	115, 122f.
Save as default configuration.....	16, 25
Save Fiducial.....	134
Save hardware settings.....	16, 149
Save payload data.....	16
Save project.....	16
Save Project.....	73
Save project as.....	16
Save project protocol.....	17
Save with options.....	16
Scale.....	91
scan head.....	143
Scanhead Selection.....	23
SCANLAB(R) RTC ScanAlone.....	51
SCANLAB(R) RTC4.....	52
SCANLAB(R) RTC5.....	53
SCANLAB(R) RTC6.....	53
Scanner.....	18
scanner bitmap.....	30, 39, 43, 47, 74, 76, 78, 85f.
scanner controller.....	143
Scanner Controller.....	18
Scanner Controller Selection.....	18
Scanner Pen Settings.....	76
SCAPS(R) USC-1/2/3.....	54
Schneider/IMS MDrive+.....	59
Second radius.....	100
Send full data to all cards.....	137
Send Named Stand Alone Data.....	137
Send Stand Alone Data.....	137
Serial interface.....	108
Serial numbers.....	139
serial port.....	110
Serial port output.....	110
Serial/Date/Time.....	117
Set Defaults for Bitmap-Marking.....	78, 85
Set frequency before MO.....	32, 36, 41, 44, 48
Set invisible.....	93
Set monospaced.....	105
Show 3D Mesh.....	131
Show All Slices.....	128f.
Show State.....	137
Show Tab-Panes in Mark-Dialogue.....	24
Sill(R) Focus Ethernet.....	67
Sine Distortion.....	115
Single arrows.....	94, 139
Sintec(R) ETH6608.....	55
Skip fixed size.....	96
Skip marks.....	93

Skip offset.....	94
Sky-writing.....	82f.
Sky-Writing.....	82
Slice Direction.....	126f.
Slice Distance.....	125, 127
Slice during marking.....	128
Slice into layers.....	125
Slice Offset.....	125, 127
Slope.....	25, 101
Smart Factory.....	72, 181
Smart Interface.....	72, 93, 181
Smartphone.....	181
Smoothing Factor.....	101
Snap to Grid.....	18
SP-ICE2.....	50
Spatial correction X.....	74
Spatial correction Y.....	74
Speed.....	94, 139
SPI G4 Laser RS232 Controller.....	64
Spikes.....	101
spiral.....	25
Spiral.....	101
Split.....	123
Split to tiles.....	126
Split To Tiles.....	113, 129
split view.....	90
spot size.....	113
Spot Size.....	75
Standard outline.....	104
Star.....	100
Start Angle.....	100, 102
Start Video.....	133
Status Message.....	182, 186
Step pulsewidth.....	58
Stepper Motor Driver (Scannercard).....	60
STL.....	125
Stop axis.....	107
Stop image capture during marking operations.....	23
Stop image capture on motion operations.....	23
Supervision.....	73
Supervisor.....	73
Support Pens.....	131
Support structure model.....	131
Support Structures.....	130
Suppress MOTF.....	138
Surface Tessellation Language.....	125
Sweep.....	78
Switch back to old OpenGL.....	71
<b>T</b>	
Teach Fiducial.....	134
Text.....	103, 115
Tile drop offset.....	114
Tile Overlap.....	114

Tile Size.....	114, 126, 129f.
Tile Skip Size.....	114, 126, 129f.
time lag.....	84
To all pens.....	75
To produce.....	139
Toggle laser.....	140
Tool.....	115
Toolbars.....	72
Top To Bottom.....	126f.
top view.....	90
Topic.....	72
touch-screen.....	198
Trace data.....	25, 93, 183
Trace log.....	25, 93
traceability.....	93, 183
tracking error.....	84
Tracking error.....	84
TransportFinished.....	72
TransportFinished distance.....	72
Triangle.....	98
TriggerUI.....	156, 178
TrueType.....	103f.
Try split between pieces.....	121
Turn off MO during jumps.....	32, 36, 41, 44, 48
Turn off power during jumps.....	31, 36, 40, 43, 48

## U

ucf.....	55
UnGroup.....	120
UnGroup elements.....	95
Uni.....	160
Unicode.....	103
Use as coloured scanner bitmap.....	86
Use as default Pen.....	75
Use as ground shape for working area.....	125
Use for Pilot-Laser.....	75
Use GRBL syntax.....	28
Use individual elements outline.....	138
Use parameters for pen.....	76
Use pen from 3D mesh.....	127
Use preview-projector.....	71
Use real element shapes.....	138
Use spotsize from pen.....	113
Use total projects outline.....	138
User interface.....	18, 71
user management.....	198
Users.....	73

## V

variable polygon delay.....	77
Video Controller.....	133
video projector.....	71
Visual grid size.....	18

## W

Wait for external trigger.....	138, 144
Wait for tool.....	21
WaveFront.....	125
Wobble Amplitude.....	78
Wobble Frequency.....	78
Wobble mode.....	78
Wobble update.....	78
working area.....	18
Working area.....	18
Working Area Size.....	189
Write Stand Alone Data.....	137
WriteLogNote.....	156
X	
X-angle.....	189
XML.....	181
Y	
Y-angle.....	189
Z	
Z-Distance.....	189
Z-position for jumps.....	28, 35
Z-Shifter.....	23, 110
Zig-zag.....	112
Zoom.....	74
Zoom In.....	89
Zoom Out.....	89
Zoom To Selected.....	89
Zoom To Working Area.....	89
.....	19f., 22, 31f., 36f., 40f., 43ff., 48f., 114, 170, 173f., 176