Photonic Professional (GT)

User Manual

23 October 2017

Nanoscribe GmbH
Hermann-von-Helmholtz-Platz 1
76344 Eggenstein-Leopoldshafen
Germany

web: www.nanoscribe.com
e-mail: info@nanoscribe.com
phone: +49 721 981 980 0

The Photonic Professional (GT) laser lithography system is subject to changes without prior notice.

Release Date: 23 October 2017
Revision: 536

Contents

1. Technical Parameters

1.1. Electrical properties

mains voltage	AC 100 V – 240 V
mains frequency	50/60 Hz
maximum current	5 A
mains supply overvoltage	Category II
grounding equipment conductor	required
electric safety	according to DIN EN 61010-1:2010

1.2. Ambient conditions

operating conditions	indoors
operating temperature	22°C (± 4°C)
temperature stability	< 1K overall fluctuation
storage temperature	20°C (± 20°C)
relative humidity	maximum 60 %
altitude	max. 2000 m
vibration isolation	required (vibration isolation table included in PPGT)
light	yellow (A filter which blocks everything below <589 nm)

1.3. Weights and measures

total weight 250 kg	
dimensions (working station)	120 cm × 120 cm × 60 cm
weight (working station)	180 kg
dimensions (rack)	56 cm × 60 cm × 64 cm
weight (rack)	50 kg
required space (fan and working space)	200 cm × 300 cm × 200 cm
required distance to walls	30 cm

1.4. Photonic Professional (GT) specifications

lateral feature size[1]	200 nm
lateral resolution	500 nm
piezo range	300 µm × 300 µm × 300 µm

[1]lateral linewidth in IP-L 780 with a 63x objective for a so-called woodpile structure

1.5. Laser parameters

system laser class	1 (IEC 60825-1:2014)
internal laser class	3B (IEC 60825-1:2014)
center wavelength	780 nm
average output power	$<$180 mW
peak power	25 kW
pulse duration	$\approx$ 100 fs
repetition Rate	80 MHz

1.6. Intended use

The Photonic Professional (GT) is intended to be used as lithography system to manufacture 2D, 2.5D and 3D polymer structures with feature sizes from sub-micron to the millimeter scale.

2. Safety Instructions and Initial Operation

2.1. Safety instructions

Prior to using the Photonic Professional (GT) make sure to carefully read this user manual. During the commissioning and training the Nanoscribe engineers will provide the basic knowledge about the safe use of the Photonic Professional (GT). Both, safety awareness and systematic reading of this manual ensures safe work with your Photonic Professional (GT). The following safety precautions are to be read, understood and applied by anyone working with the lithography system.

Note
The Photonic Professional (GT) complies with the federal performance standard as required by EN 61010-1:2010 and IEC 60825-1:2014.

Warning!
If the equipment is used in a manner not specified by Nanoscribe, the protection provided by the equipment may be impaired.

Make sure to carefully read the manufacturers' user instructions for all components of your Photonic Professional (GT). Following manufacturers' manuals are provided with the system:

- Axio Observer and Definite Focus user manual (microscope)
- Piezo controller (E-761) user manual
- Laser user manual
- Motorized stage printed user manual or PDF-file.

2.2. Type logo

Please note the ratings stated in chapter 1 and printed on the type logo, attached to the left side of the electronics rack. This logo is also depicted in figure 2.1.

Note
Please be aware of the listed ratings before initial operation of the system.

Note
Make sure to carefully read the user manual completely before operating the system.

Figure 2.1.: Type logo of the Photonic Professional (GT)

2.3. Electric safety

Danger!
Procedures other than those specified in this documentation or use of controls deviating from the instructions in this manual may result in endangerment of electric shock.

Caution!
Performance of procedures other than those specified in this documentation or use of controls deviating from the instructions in this manual may result in damaging highly sensitive devices by electrostatic discharge.

Make sure to comply with the following safety instructions throughout the whole operation of the system:

- Ensure that the environment meets the required ambient conditions denoted in this manual.
- Store, install and operate the system solely in a clean and dry environment.
- All plugs must be inserted firmly and securely prior to operation.
- Ensure that cables are not close to sharp-edged and/or hot objects. Assure that the insulation stays undamaged.

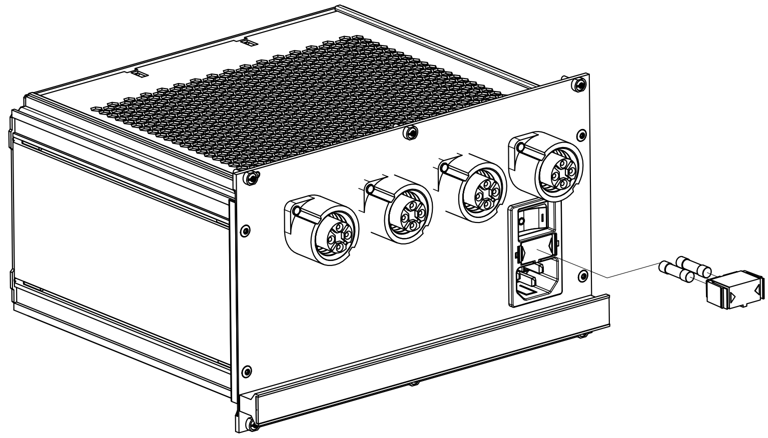


Figure 2.2.: Exchange of fuses on the Photonic Professional Controller

Disconnecting Device
The main power plug is the system's disconnecting device. The power socket to which the system is connected and the main power plug thus have to be accessible at any time!

Mains supply cords
Do not replace the mains supply cords by inadequately rated cords. Make sure that your power plug is equipped with a protective earth conductor!

2.3.1. Accessible fuses

The Photonic Professional (GT) has two main fuses in the Photonic Professional Controller. For accessing the fuse please power down the system as described in 2.8.2 and open the back panel of the electronics rack. The fuses are located in the back of the Photonic Professional Controller. Figure 2.2 illustrates how to exchange the fuse. Please make sure to use fuses of this type: 5 A T

If one of the fuses blows, the cause should be identified and any technical defect remedied before replacing the fuse. Prior to changing any fuse make sure that the Photonic Professional Controller is turned off and that the mains power plug is disconnected. Open the spring loaded fuse carrier. Replace the defective fuse and reinsert the carrier as shown in figure 2.2.

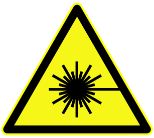
The laser controller, the microscope power supply and the galvo power supply have additional fuses. Please refer to the respective manufacturers' user manuals for a detailed description of the exchange of these fuses.

2.4. Laser safety

2.4.1. Laser source and protective housing

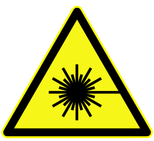
The Photonic Professional (GT) is classified to laser protection class 1 (IEC 60825-1:2014). The system is equipped with a class 3B femtosecond laser (see 1.5). The protective housing

prevents the direct contact with the laser radiation. The laser and the laser beam are inaccessible to the operator during standard operation of the system.



Warning!
Never remove any parts of the protective housing of the Photonic Professional (GT)! In case of damages or doubts about completeness of the protective housing please switch off the system according to the procedure described in 2.8.2 immediately and contact your Nanoscribe service representative for further instructions.

During commissioning and servicing the protective housing may need to be removed by the Nanoscribe service engineer for laser adjustment reasons. During these periods the Photonic Professional (GT) is classified equivalent to the equipped laser as laser protection class 3B with the parameters given in chapter 1.5. Make sure that your laboratory is equipped with all safety requirements for running a laser of class 3B prior to the installation of the Photonic Professional (GT). For additional information on the laser parameters, specifications and safety measures please refer to the user manual of the laser manufacturer which is provided along with this user manual.



Warning!
Procedures other than those specified in this documentation or use of controls divergent from the instructions in this manual may result in hazardous radiation exposure. The laser source of the Photonic Professional (GT) is a class 3B laser device and hazardous for eyes and skin.

2.4.2. Illumination and autofocus system

The Photonic Professional (GT) is equipped with two LEDs at 625nm for the visualization of the printing process. These two LEDs are rated class 1 after ISO 62471:2009, which means an exposure for less than 10s is uncritical. Please avoid staring in to the activated LEDs!

2.4.3. Interlocks

The Photonic Professional (GT) is equipped with one or two optional electric interlock switches (model dependent). Note that both interlocks are not rated according to the relevant laser safety norms. The interlocks are optional (not required but standard in the system) and will not be subject of any use in any standard operation process of the Photonic Professional (GT). The cabinet interlock is located in the rear of the optics cabinet and shuts down the laser immediately when the optics cabinet is opened. The stage interlock is mounted on the stage lid. As soon as the lid is opened the laser is shut down.
The system may be connected to a remote interlock via a M8 3 pin interlock connector in the rear panel of the Photonic Professional Controller

2.4.4. Shutters

The Photonic Professional (GT) is equipped with an electrical shutter which blocks the laser beam path towards the microscope as long as no exposure job is running. Note that the shutter is not rated according to the relevant laser safety norms.

2.4.5. Indicators

The interlock/shutter status of the Photonic Professional (GT) is indicated via the LEDs on the front panel of the Photonic Professional (GT) controller and on the door of the electronic rack. The interlock LED lightens up as soon as an interlock is opened. The shutter LED lightens up while the shutter is closed.
During standard operation all interlocks are closed and the electric shutter alternates between closed and open. The interlock indicators are thus inactive, whereas the shutter indicators may change between on and off. During the execution of writing jobs the shutter is opened most of the time, thus, both LEDs are inactive most of the time.

2.5. Operation safety

2.5.1. Substrates

The Photonic Professional (GT) is intended to be used with the dedicated glass substrates or with silicon wafers. Please make sure to clean the sample holder thoroughly in case the substrate breaks. Please take care not to cut yourself on the glass or silicon pieces. We recommend to use plastic tipped tweezers to remove all glass or silicon pieces.

2.5.2. Objective revolver and microscope z-drive

Please note that the z-drive as well as the objective revolver are motorized components. Take care when handling in proximity to those components not to bruise your fingers or hands due to sudden movements of these components.

2.5.3. Materials

The Photonic Professional (GT) may not be operated with any flammable or explosive material at all, this includes but is not limited to chemicals and gases. Electrical voltages, (focused) LED and laser radiation are possible sources of ignition.

2.5.4. Motorized stage

The mechanical safety of the Photonic Professional (GT) focuses on the motorized stage. Make sure to remove any obstacles before you start working with the Photonic Professional (GT). Never touch the motorized stage while moving.

Danger!
The movements of the motorized stage is in most cases automated and no emergency stop is implemented. Make sure to avoid obstacles in the movement range. Note the warning label attached to the motorized stage.

2.5.5. Optical table

The breadboard is pneumatically leveled and lifted by about 1 cm. Please take care not to get any part of your body inbetween the breaboard and the 4 legs of the optical table base.

2.6. Warning labels

The Photonic Professional (GT) is equipped with all relevant warning labels. Two different laser warning labels shown in figure 2.3 are situated at the following locations on the system: both sides of the optics cabinet , the piezo front and the stage lid. Furthermore a mechanical hazard warning label is located on the motorized stage as shown in figure 2.4.

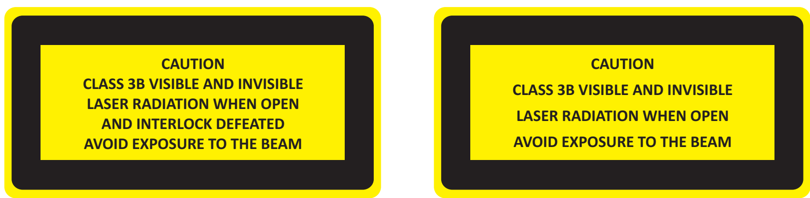


Figure 2.3.: The two types of laser warning labels are attached to the system

Figure 2.4.: Warning label attached to the stage

2.7. Initial operation

The following chapter covers how the Photonic Professional (GT) is packaged, what is supplied with the Photonic Professional (GT) and the requirements for setting-up the Photonic Professional (GT). Together with the chapter 2.8 it briefly explains all the necessary steps of an initial operation of the Photonic Professional (GT). However, this description is not intended as a guide to set-up your Photonic Professional (GT) yourself. This task is assigned to a Nanoscribe service engineer as part of the sales agreement.

A Nanoscribe service engineer will contact you to arrange the commissioning of your Photonic Professional (GT). Allow only specialists qualified and authorized by Nanoscribe to install and commission your lithography system. We do not guarantee the proper operation and safety unless the Photonic Professional (GT) is installed by a member of the technical staff of Nanoscribe or an authorized representative of Nanoscribe.

Study the entire manual before putting the Photonic Professional (GT) into operation. Make sure you are familiar with all aspects of the installation and operation before handling the equipment. Pay particular attention to the safety instructions within in this user manual.

Warning!
Do not attempt to install the Photonic Professional (GT) yourself! Unauthorized installation will void the warranty! You might be charged for any damage incurred if you attempt to install the system yourself!

2.7.1. Packing

The Photonic Professional systems are either shipped or directly transported by our service team to its designated installation location. If the system is shipped, it comes packed either in two (Photonic Professional) or three (Photonic Professional (GT)) wooden transport crates:

Photonic Professional (GT):

Crate 1	dimensions:	128 cm x 118 cm x 110 cm (l x w x h)
	weight:	370 kg
Crate 2	dimensions:	118 cm x 138 cm x 115 cm (l x w x h)
	weight:	360 kg

Photonic Professional (GT):

Crate 1	dimensions:	128 cm x 118 cm x 110 cm (l x w x h)
	weight:	340 kg
Crate 2	dimensions:	118 cm x 138 cm x 115 cm (l x w x h)
	weight:	300 kg
Crate 3	dimensions:	118 cm x 138 cm x 115 cm (l x w x h)
	weight:	300 kg

After receipt of the transport containers we recommend to store the containers in the location the system will be operated. The highly sensitive components require some time to adapt to the environmental conditions. In case the containers have to be stored in another location make sure to choose a location which is dry and ideally temperature controlled (15-25 °). Please do not unpack the Photonic Professional (GT) from its transport crates. This task is assigned to a Nanoscribe engineer as part of the sales agreement. Please only allow persons qualified and authorized by Nanoscribe to install and commission your Photonic Professional (GT).

Warning!
The Photonic Professional (GT) comprises highly sensitive components. Mechanical agitation, electrostatic discharge or rapidly changing environmental conditions might cause damage to the unit or downgrade its performance significantly.

2.7.2. Scope of Supply

The following items are part of the delivery of the Photonic Professional (GT):

- optics cabinet on breadboard including the laser
- microscope with positioning stage
- optical table base frame
- electronics rack with computer and controllers
- monitor, keyboard and mouse
- stage joystick and microscope docking station
- software
- accessory (cables, substrates, etc.)
- user manual laser
- user manual microscope
- user manual piezo controller board
- user manual motorized stage (CD)
- user manual Photonic Professional (GT)
- GT-option (optional)
- sample holder(s) (optional)
- microscope objective(s) (optional)
- working desk (stainless steel or wood) (optional)

2.7.3. Requirements

Several arrangements and some planning are required before installation of the Photonic Professional (GT).

- Select a suitable location for the Photonic Professional (GT).
- The environment has to meet the specifications described in 1.2.
- The laboratory must not be exposed to daylight and needs amber light.

- Make available sufficient utilities and diagnostic equipment for the pre- and post-processing of the structures.
- Ensure that the designated location is clear and accessible for heavy equipment.

Infrastructure

Recommendable is a room that is free from dust and that does not exhibit any nameable temperature fluctuations. Although the system is designed to minimize susceptibility to environmental effects the performance can depend on ambient conditions. It is recommended that the temperature should be controlled within ±1 K to achieve the maximum performance for your Photonic Professional (GT). Avoid any acoustic noise and mechanic vibrations around the installation location of the system.

Note
The Photonic Professional (GT) needs to be located in a laboratory environment.

We strongly recommend to mount the system on an optical table base. The necessity for vibration absorption depends on the scope of application and environmental conditions. In general this table base comes within the scope of supply.

The Photonic Professional (GT) requires a total footprint of about 2.85 m × 2 m. The customer is responsible for determining the most suitable location for the Photonic Professional (GT). We strongly recommend a well accessible location to simplify installation and maintenance procedures.

If you seek advice on the installation location and on your lab conditions please do not hesitate to contact Nanoscribe service.

Recommended Equipment

All required equipment for the installation, such as tools, measuring instruments, diagnostic systems and auxiliary materials, are brought along by the Nanoscribe engineers. Nanoscribe will provide a detailed checklist prior to the installation, with all necessary equipment for commissioning.

Caution!
The breadboard and the electronics rack are heavy weight products.
A fork lift or similar equipment is recommended for lifting those parts.

2.7.4. Connections

The Photonic Professional (GT) components are connected according to the scheme of table 2.1 and figures 2.5-2.9 by the Nanoscribe engineer during commissioning.

Table 2.1.: Cable connections of all system components

computer

1.01	→	4.02 Microscope backpanel
1.02	→	Monitor DVI
1.03	→	6.01 AF Camera
1.04	↔	LAN 1
1.05	→	2.11 Signal IO 0
1.06	→	7.01 Axio camera
1.07	→	5.01, 5.02 Motor Stage
1.08	→	5.03 Cl x
1.09	→	5.03 Cl y
1.10	←	Piezo scanner
1.11	→	10.02 Laser controller
1.12	←	Mouse
1.13	←	Keyboard
1.14	→	USB Monitor
1.15	→	USB Rack
1.16	→	2.12 Signal IO 1
1.17	←	Joystick
1.18	→	2.13 Modulation
1.19	←	2.10 AC power in

Photonic Professional (GT) controller

2.01	→	Galvo x
2.02	→	Galvo y
2.03	→	3.01 Optics cabinet
2.04	←	Reflection illumination
2.05	←	Transmission illumination
2.06	→	6.02 AF power
2.07	→	12.01 AC out
2.08	→	11.01 AC out
2.09	→	10.01 AC out
2.10	→	1.19 AC out
2.11	←	1.05 Signal IO 0
2.12	←	1.16 Signal IO 1
2.13	→	1.18 Modulation
2.14	→	Rack
2.15	←	Interlock
2.16	→	10.03 Laser
2.17	←	Stage interlock
2.18	→	Protective earth
2.19	→	Protective earth
2.20	←	AC power in

optics cabinet

3.01	→	2.03
3.02	→	10.04

microscope

4.01	→	8.01
4.02	→	1.07
4.03	←	Microscope powersupply

xy-stage

5.01	→	1.07
5.02	→	1.07
5.03	→	1.08, 1.09

microscope autofocus

6.01	→	1.03
6.02	→	2.06

live-cam

7.01	→	1.06

microscope autofocus

8.01	→	5.03

Laser controller

10.01	→	2.09
10.02	→	1.11
10.03	→	2.16
10.04	→	3.02

microscope power supply

11.01	→	2.08

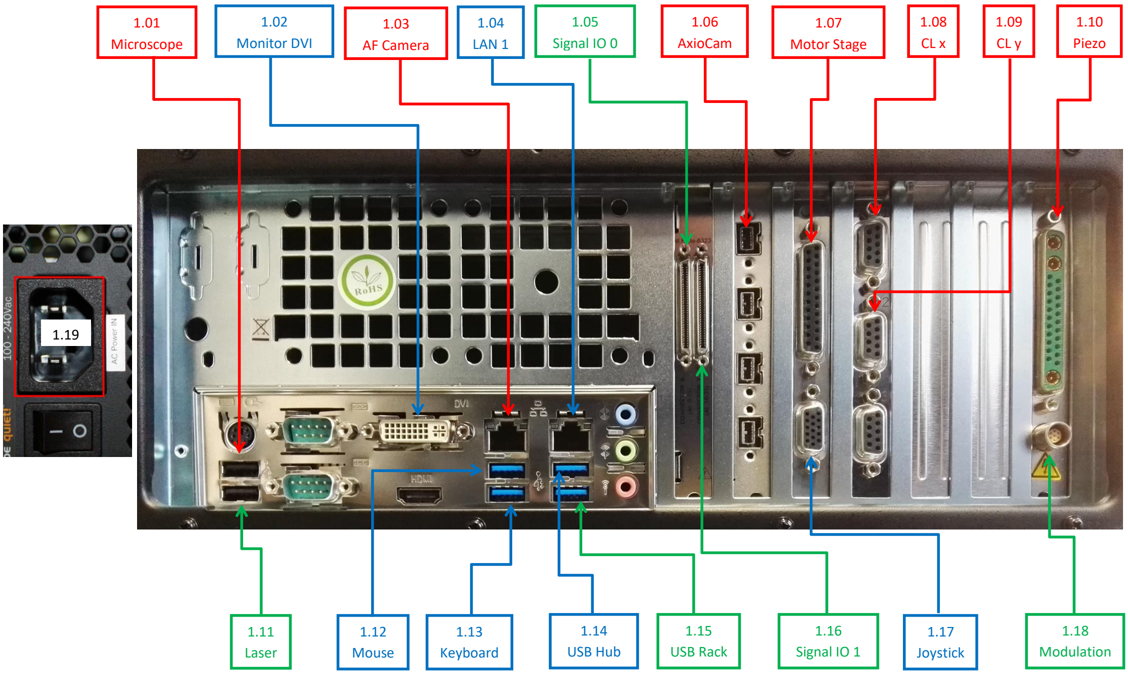


Figure 2.5.: Computer rear side

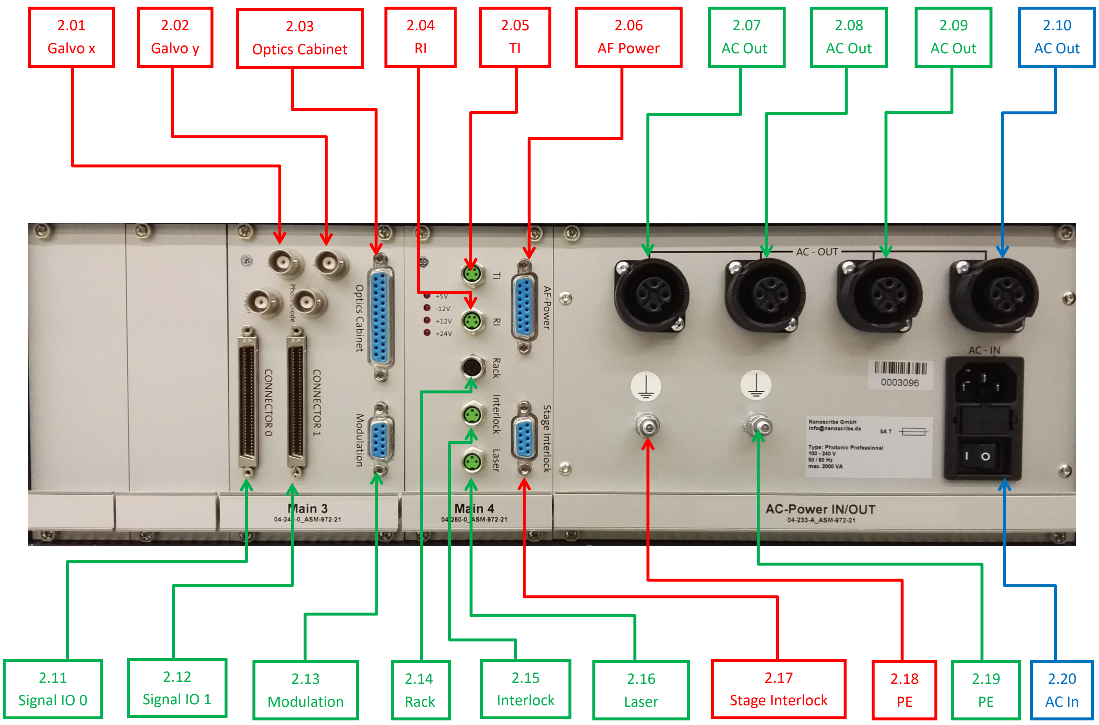


Figure 2.6.: Electronics box rear side

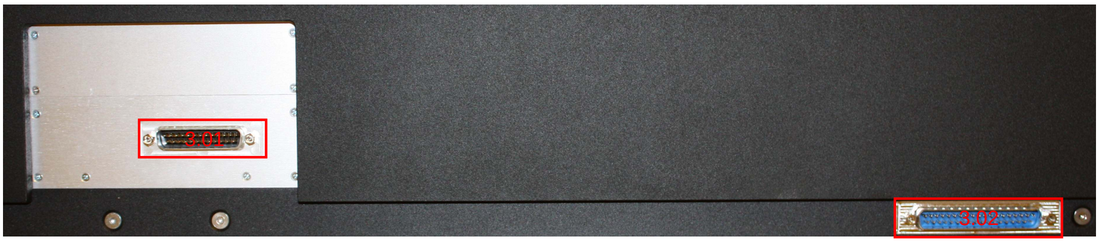


Figure 2.7.: Optics cabinet rear side

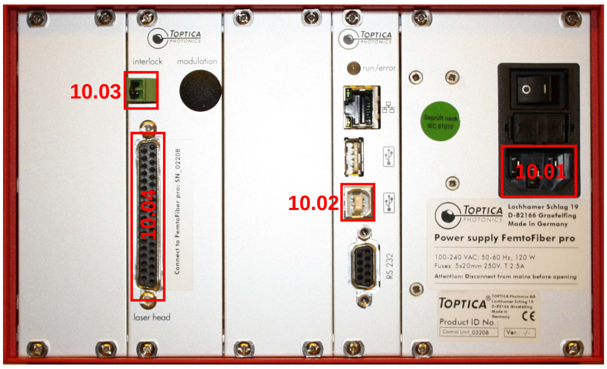


Figure 2.8.: Lasercontroller rear side

Figure 2.9.: Autofocus (6.), live camera (7.), motorized stage (5.), microscope backpanel (4.), microscope docking station (8.) and microscope power supply (11.)

2.8. Power-up and power-down procedure

This is a summary of the steps to be performed for a proper power-up and power-down of the Photonic Professional (GT). Please note that in some cases your configuration might differ slightly. However, under such circumstances it should be straightforward to adapt to your specific case. If in doubt, please contact your device administrator or Nanoscribe service. For identification of the individual components please refer to chapter 3. For Laser specific details please refer to chapter 2.4, 3.5 or the respective device manual. Regarding the operation of the particular component refer to the respective documentation or user manual for details.

2.8.1. Power-up the Photonic Professional (GT)

Warning!
Disregard of the guidelines concerning power-up of the Photonic Professional (GT) may cause damage to the system or downgrade its performance significantly.

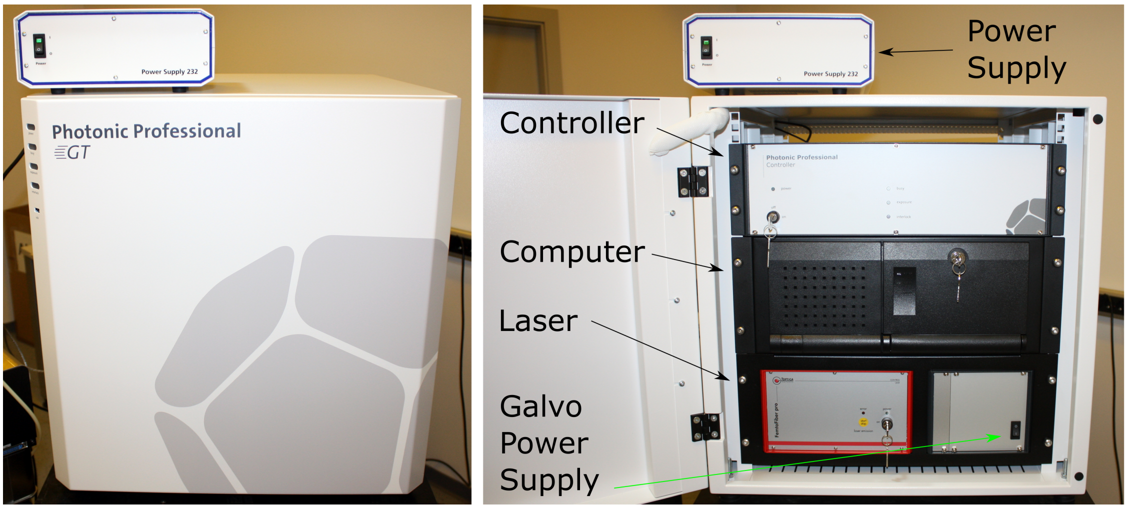


Figure 2.10.: Closed (left) and open (right) electronic rack of the Photonic Professional (GT).

Switch on the components in the following order in order to power-up the Photonic Professional (GT) (Fig. 2.10):

1. Photonic Professional (GT) Controller by activating the key switch.
2. Computer: Let it boot completely before proceeding, wait until the log on screen shows up.
3. Laser controller by activating the key switch. The button 'start/stop' does not have to be pushed (the backlight of the button should be off). The laser emission is controlled through the software NanoWrite.
4. Power supply of the galvo (if GT-system).
5. Power supply unit of the microscope.

6. Microscope by pressing the silver button on its left side.
7. Monitor.
8. Start NanoWrite

The laser gives an acoustic feedback, as soon as its start up is finished. This can take up to 30 s. After that the control software NanoWrite may be launched. Make sure that the laser is operational when starting NanoWrite. The laser emission is switched on automatically when NanoWrite starts. The status of laser emission is indicated by the backlight of the button 'start/stop' (7) on the front panel of the laser controller. If the light is activated, the emission is on. You can manually switch on/off the emission by pressing this button. A power calibration of the laser is performed automatically before the first print starts or when NanoWrite receives the first script command.

Warning!
Make sure the objective lens is completely lowered, before inserting a sample holder, to avoid damage to the objective lens. The message 'lower z-limit reached' should be shown on the microscope touch panel.

Note
Please keep in mind that all components need a certain time to evolve to their thermal equilibrium. Only then they will operate ideally within their specifications. Therefore, we recommend to wait for an adequate time, about half an hour, before starting the first job. For further information about running-in periods of individual components, please refer to the respective documentation or user manual.

2.8.2. Power-down Procedure

Warning!
Disregard of the guidelines concerning power-down of the Photonic Professional (GT) may cause damage to the system or downgrade its performance significantly.

To power-down the Photonic Professional (GT), follow this order (Fig. 2.10):

1. Exit NanoWrite. Wait till NanoWrite is shut-down. Otherwise, error messages may occur. In general the laser emission is switched off automatically when exiting NanoWrite.
2. Shut-down the computer.
3. Make sure the microscope objective lens is in the lowermost position. Check for the message 'lower z-limit reached' on the docking station of the microscope.
4. Switch off the microscope (silver button on its left hand side).

5. Switch off power supply of the microscope.
6. Switch off power supply of the galvo.
7. Check the status of the laser emission; if the backlight of the button 'start/stop' is activated, press it to switch off the emission.
8. Switch off the laser controller by deactivating the key switch.
9. Switch off the Photonic Professional (GT) Controller by deactivating the key switch.

Warning!
Never switch off the Photonic Professional (GT) Controller prior to any of the other components. This might cause damage to the system or downgrade its performance.

2.9. Troubleshooting

Observation	**Action**
One of the system components does not power up.	Inspect the respective connections 2.08 - 2.11 on the controller and the corresponding power plug on the component.
The Photonic Professional (GT) Controller does not power up.	Switch off the controller. Then carefully inspect whether the main power plug is well connected to a power socket and retry.
One or more components are not correctly recognized by the computer, error messages are generated when starting NanoWrite.	Restart the computer and check if the error messages are still generated.
An error LED enlightened.	Power down the system, restart it completely and recheck the error LED.
The interlock LED on the Photonic Professional (GT) Controller is lightening up.	Check thoroughly if all interlocks on the system are closed. Check the remote interlock connection.
The microscope is displaying a different z-drive position (hundreds of micrometers or more) on its touch panel than in NanoWrite.	Close NanoWrite, click 'load position' on the microscope touch panel, switch off the microscope, switch on the microscope and restart NanoWrite. The displayed position should now coincide plus/minus a few tens of micrometers.
There is no signal on the monitor.	Switch on the monitor by bushing the 'on' button.

3. Hardware

This chapter presents the standard hardware and accessory of the Photonic Professional (GT). The focus lies on introducing the naming conventions used throughout this manual. Detailed explanations on the usage of each component and all procedures necessary for proper operation of the system are introduced in the following chapters. Figure 3.1 shows a complete system standard setup with indications of the major components.

Warning!
The Photonic Professional (GT) is a stationary unit. Customers may not move the system on their own. If a relocation should become necessary please contact us (service@nanoscribe.com).

3.1. Optical table

The system is installed on an optical breadboard which is passively vibration isolated by a self leveling optical table base frame. The optical table frame has to be supplied with about 5 bar compressed air or nitrogen to ensure optimal vibration damping. A 1/4 inch outer diameter hose is shipped with the system which has to be plugged to the compressed air supply of the installation location. The overall specs of the optical table are:

dimensions:	110 cm × 90 cm × 90 cm
weight:	180 kg

Nanoscribe supplies special cleanroom compatible optical table base frames. Please contact your Nanoscribe sales representative (sales@nanoscribe.com) for further information on this option.

3.2. Optics cabinet

The laser and the optics are mounted on the breadboard and covered by a protective housing, generally referred to as the optics cabinet.

Warning!
Do not unscrew or open any parts of the protective housing of the Photonic Professional (GT).

3.3. Microscope and autofocus system

The system comprises a Carl Zeiss Axio Observer microscope for focusing the laser beam into the photoresist and for observation of the printing process. The microscope is equipped with the following components (see figure 3.2):

- live camera
- focus dial

Figure 3.1.: Overview of the Photonic Professional (GT).

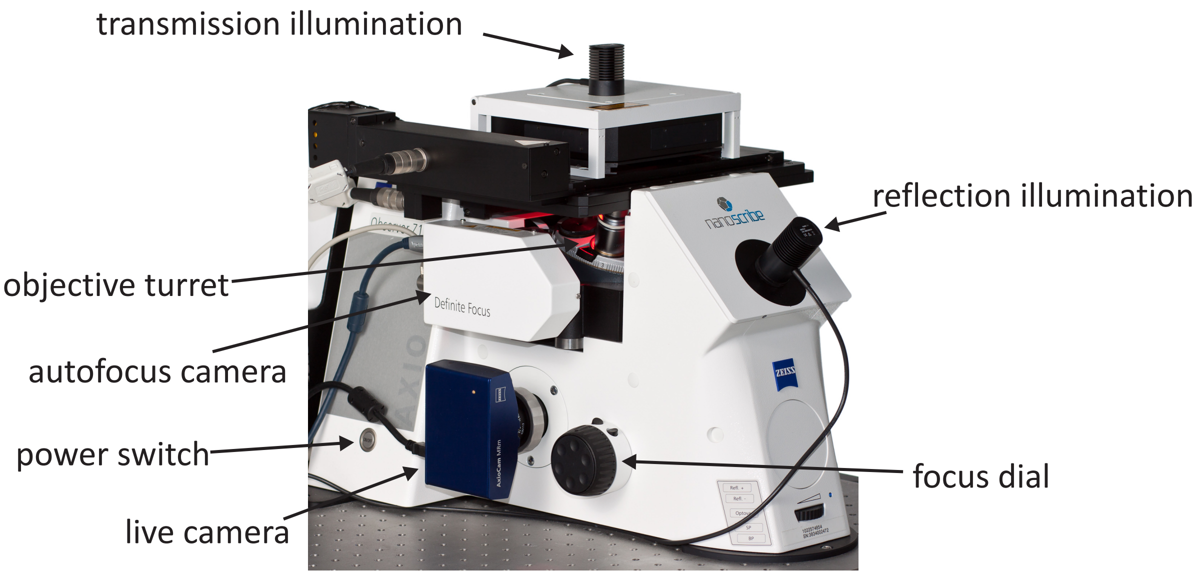


Figure 3.2.: Microscope with major components

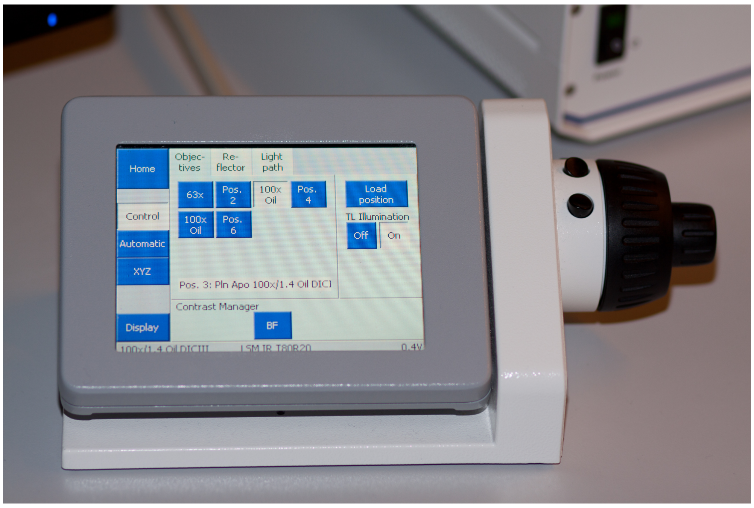


Figure 3.3.: Microscope touch screen display in docking station

- autofocus system
- objective turret
- transmission illumination
- reflection illumination

The microscope is operated via a touch screen display in docking station as shown in figure 3.3. All relevant parameters may be set using this docking station. Please refer to the user manual of the microscope for more details about the adjustments via the docking station.
There are two illumination options. A LED for transmission illumination and a LED for reflection illumination.

The microscope is equipped with an autofocus system to automatically focus the laser spot at the substrate surface. The autofocus system is working via an imaging approach of a grating on the front focal plane of the microscope objective. The position of the image of the grating on the autofocus camera is an indicator of the position of the focal spot of the laser on the substrate. A detailed description of the autofocus system may be found in the manual of the Definite Focus.

3.4. Piezo and motorized stage

The Photonic Professional (GT) comes with a motorized stage for coarse positioning and a piezo for fine positioning. The stage automatically addresses different substrates or different writing positions on one single substrate. The stage coordinate system is absolute on each sample with the stage origin (SO) in the sample center. The accessible writing area on each substrate is indicated in figure 3.4. The exact dimensions are specified in the file exchanger.ini in the NanoWrite directory.
The piezo is mounted on top of the stage and is thus displaced by the stage movements. The piezo coordinate system is 300x300x300 μm^3 with its origin (PO) at the current stage position. No negative coordinates can be addressed by the piezo.

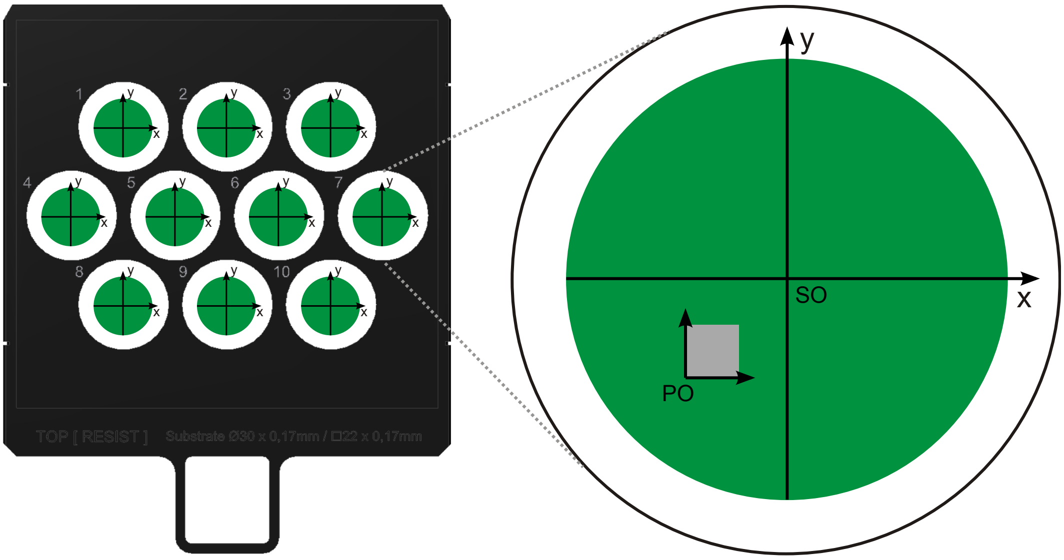


Figure 3.4.: Stage and piezo coordinate system. Note that the indicated directions are writing directions (viewed from top). They represent a right handed coordinate system, whereas the movement of the stage and piezo are left handed.

Hardware Danger!
Avoid driving the piezo at its resonant frequency (about 100 Hz) to avoid damages. Check your designs and make sure no periodically back and forth movements close to 100 Hz or higher are programmed. Notice that the structure size and the writing speed as well as the actual structure design play a crucial role. Furthermore the writing mode and the SettlingTime have to be considered.

3.5. Laser

The Photonic Professional (GT) is equipped with a pulsed femto second fiber laser source at a center wavelength of 780 nm. The laser power ranges between 50 mW and 150 mW at a pulse length between 100 fs and 200 fs. The lasers are turnkey systems which are completely integrated to the Photonic Professional (GT). For the standard procedures the laser parameters and settings should not be changed in any way by the user of the system. No powering up and powering down is needed for standard procedures. The lasers may run continuously. For further instructions on the usage and safety of the lasers please refer to the respective user manual.

3.6. Electronics rack

The electronics rack is in general located beneath the optical table. The electronics rack houses the Photonic Professional (GT) controller, the computer, the laser controller and the galvo controller (for GT-systems).

4. Software

This chapter introduces the standard software applications that are commonly used in conjunction with Nanoscribe's Photonic Professional (GT) laser lithography system. However, it does not provide detailed instructions for using the applications themselves as these will be covered in later chapters of this manual. Please refer to chapter 5 on page 31 for an introductory step-by-step guide of how to use the Photonic Professional (GT).

4.1. NanoWrite

NanoWrite provides an easy-to-use graphical user interface for controlling your Photonic Professional (GT), see figure 4.1. It interfaces with and integrates the system hardware described in chapter 3. Although NanoWrite contains facilities for manual control and system calibration, its main purpose is to drive the fully-automated direct laser writing process while allowing you to easily monitor the system status and writing progress. For this purpose, NanoWrite features a built-in live display of the microscope camera.

In order to fabricate a structure with the Photonic Professional (GT), NanoWrite requires as input a file defining the structure in Nanoscribe's proprietary General Writing Language (GWL). A GWL-file describes the trajectories to be followed by the laser focus inside the photoresist and also configures all the system parameters to be used during the writing process. When you select a GWL-file to load into NanoWrite, it first checks the file for errors and compiles the GWL-files to GWLC-files for printing. If the 'display structure' button is activated, it also computes an estimate for the time required to write the structure. In this case NanoWrite also displays a freely rotatable and zoomable three-dimensional preview of the structure in its main window, after the file has been successfully loaded. Please note, that we highly recommend DeScribe for previewing purposes. After starting the exposure the writing process can be monitored in the camera view.

4.2. DeScribe

GWL-files can be edited in any text editor. We recommend the special editing software DeScribe for authoring GWL-files, see figure 4.2. Next to providing syntax highlighting and completion, it is also checks syntax, displays structures, slices STL files (see chapter 8), pre-computes compiled gwlc files and debugs. Below we will introduce you to the most important features of DeScribe.

- **GWL editing**: DeScribe is above all a GWL editor for designing structures and developing recipes for 3D lithography. The user is aided by syntax highlighting and completion of the GWL commands used by NanoWrite. A gwl reference is included and gives you a short explanation of the commands used. DeScribe also checks for syntax errors and will help you to format your document nicely (see chapter 7).
- **STL conversion**: DeScribe imports STL files and converts them into GWL scripts. The slicing algorithm greatly simplifies the fabrication of 3D structures with the Photonic Professional (GT). Instead of manually authoring the GWL-input required by

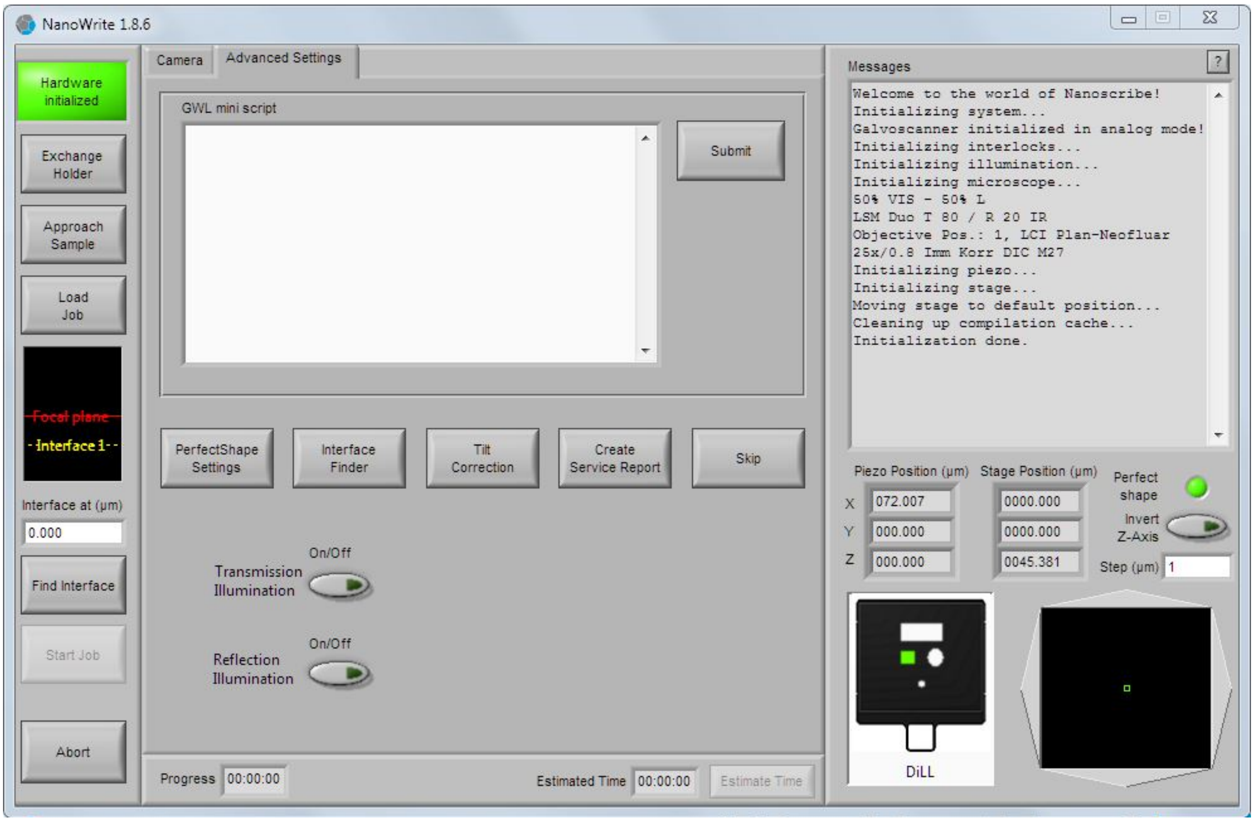


Figure 4.1.: Main application window of NanoWrite

NanoWrite, you can create a digital model of the 3D structure with a computer-aided design (CAD) application of your choosing. Almost any modern CAD application can export the digital model to the STL format. DeScribe converts this STL file into a GWL-file ready to use in NanoWrite. This conversion process requires the computation of a sequence of trajectories based on a layer-by-layer approach. Different mechanisms are available to suit your specific needs. Since the conversion process can require significant computing resources, it has been separated from NanoWrite in order not to interfere with the hardware control. Instead, the process can be off-loaded to a different and better equipped computer. Please refer to chapter 8 for usage instructions and an explanation of the working principle.

- **3D preview**: DeScribe offers different rendering modes for previewing your structures. In voxel mode, DeScribe renders lines and points as three dimensional objects representing a user chosen width and aspect ratio. This offers a better estimation of what your designed structures will look like after writing.

  The rendering mode can be modified in 3D Preview > Rendering mode. The voxel representation mode is highly demanding with respect to graphics card resources. Laptops equipped with low-level graphics cards or integrated GPUs may suffer from the lack of dedicated graphics memory when representing structures in this mode. As a result we advise working in simple line mode in such cases.

- **GWL Debugger**: The GWL debugger is an advanced development tool that helps locating and correcting programming mistakes in complex hand-written GWL code.

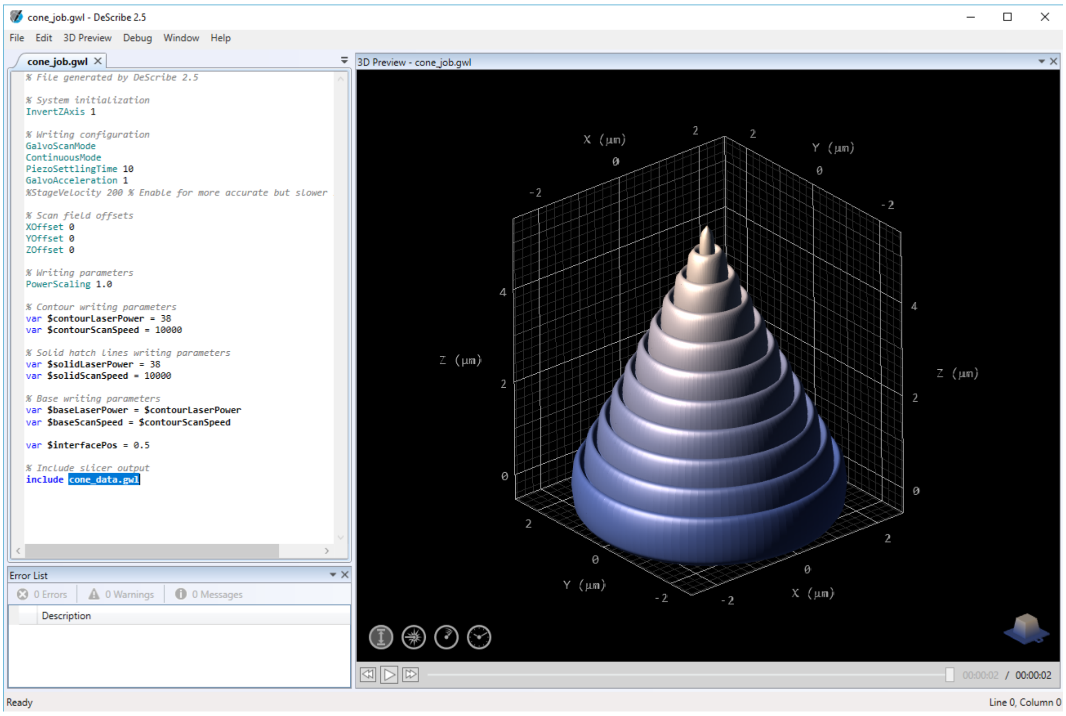


Figure 4.2.: Main application window of DeScribe

You can create breaking points, view the current buffer and system state as well as watch for variable changes.

4.3. Zeiss AxioVision

While NanoWrite includes a live display of the microscope camera, it is also possible to use ZEISS AxioVision to display the camera image. AxioVision features more configuration parameters than NanoWrite to optimize picture quality. It also allows you to resize the live picture up to fullscreen and to save a camera snapshot to disk at the native resolution of the camera.

Warning!
The live image from the microscope camera can not be displayed simultaneously in NanoWrite and AxioVision. While AxioVision is running the camera display in NanoWrite cannot be activated. But if you launch AxioVision while the camera display is active in NanoWrite, a system crash may result in some circumstances.

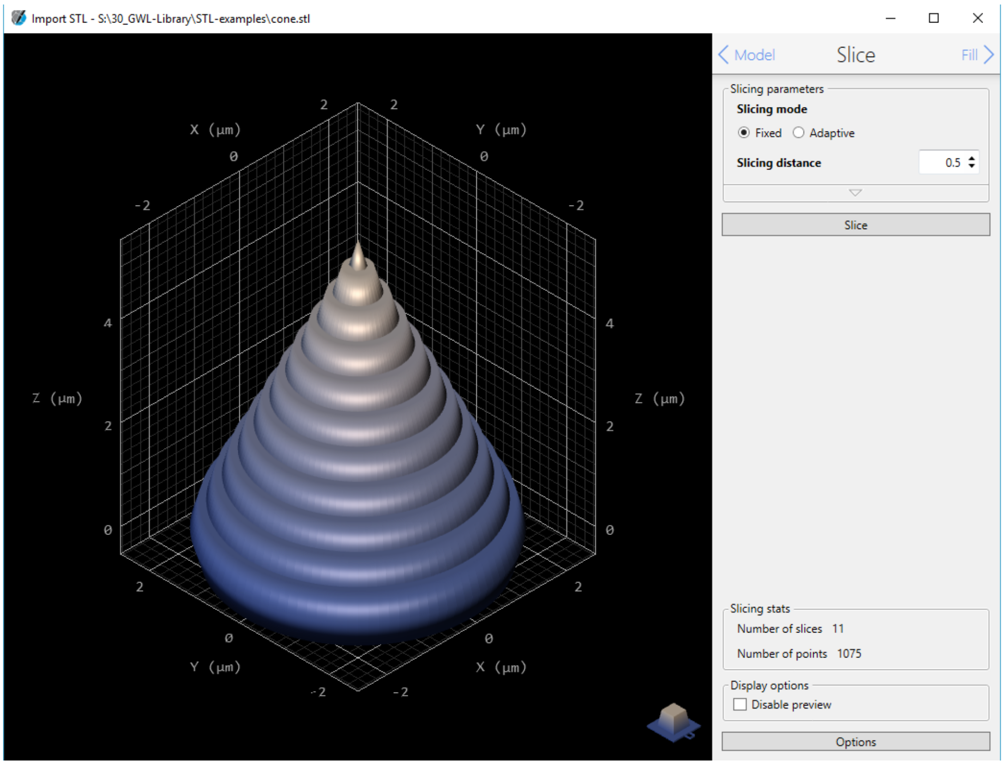


Figure 4.3.: Screenshot of the slicer assistant embedded in Describe

5. General printing workflow

5.1. Software start and initialization

When starting NanoWrite, e.g. by double clicking on the NanoWrite icon on the desktop, the system will run a initialization procedure that may take about one minute. You will see that it is finished once the top left button is lit brightly green and its label has become: “Hardware Initialized”. During the initialization the following procedures are performed:

- *Interlock Initialization*: The status of the interlocks and of the electric shutter are checked.

- *Piezo Initialization*: The connection to the piezo controller is established and the piezo axes are initialized and set to the origin.

- *Microscope Initialization*: The connection to the microscope is established and the microscope standard settings are applied.

- *Stage Initialization*: The connection to the stage controller is established and the stage is referencing its axes. A warning message is displayed that has to be confirmed for this process to complete (see Figure 5.1). In case you abort the stage calibration, the stage has to be used manually, i.e., using the joystick.

  After successful calibration the stage will move to its center position and then give the opportunity to select the desired sample holder and respective sample position. Once the stage arrived at that location, the system is fully initialized (except the laser power calibration) and ready to use.

Danger!
Make sure no obstacles are in the stage environment! The stage movement cannot be stopped and may cause serious injuries! Furthermore any cables should be well attached as installed upon commissioning.

Warning!
Make sure the microscope displays *Lower z-drive limit reached* before starting NanoWrite. If not, lower the z-drive manually until the message is displayed. You may also use the click **Load Position** button on the microscope display. If so confirm with click **Set working position** after the movement finished.

- The laser power calibration is performed when the first command is submitted to NanoWrite. This might be done either by starting a print job, by commands triggered via GWL-miniscript with click **Submit** or by any command button (e.g. click **TiltCorrection**).

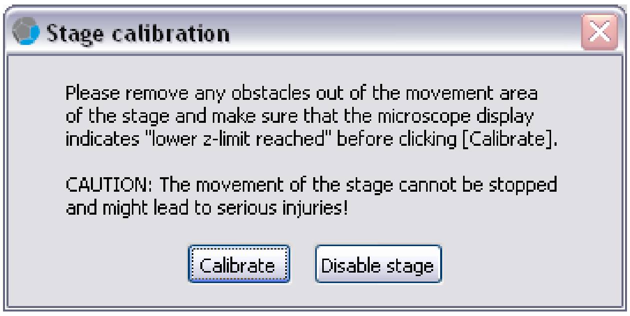


Figure 5.1.: Stage calibration warning

5.2. Printing configurations

This section introduces the three main printing configurations of the Photonic Professional (GT):

- **Oil immersion configuration:** In this configuration, also called conventional DLW, the laser beam is focused through the substrate into the photoresist. Between objective lens and substrate the immersion oil is matching the refractive index in order to get the ideal focus on the substrate-resist interface. However, the further the laser is focused into the resist meaning the higher the fabricated structure, the worse the focus quality gets due to spherical aberrations. The maximum structure height is defined by the working distance of the microscope objective and by the substrate thickness. We thus recommend this configurations where neither of the DiLL resists are applicable or where it is mandatory to focus through the substrate. See Table 5.1 for the standard hardware used for the oil immersion configuration.

- **Dip-in Laser Lithography (DiLL):** In this configuration the microscope objective is directly dipped into the photoresist. The spherical aberrations are minimized and constant for the full printing range. No interfaces limit the structure height in this configuration, the maximum structure height is limited only by the sample holder used and may be larger than 2 mm. We recommend to only use the index matched DiLL compatible photoresists of Nanoscribe for this printing technique with the respective compatible objective lenses. Other resists might damage the objectives or result in bad printing quality due to a non ideal index match. In Figure 5.2 the oil immersion and DiLL configuration are compared graphically. Please note that the writing direction in DiLL configuration is upside down. Either use the command `InvertZAxis 1` in your script to set the correct writing direction or activate the **Invert Z-Axis** button in NanoWrite. The piezo x- and z-axes will move to 300 µm, which corresponds to the inverted piezo origin, to conserve the right-handedness of the coordinate system and flip the writing direction at the same time. See Table 5.2 for the standard hardware used for the oil immersion configuration.sample approach

- **Air configuration:** In this configuration the immersion medium is air. The laser can be focused either through the glass substrate, or directly into the photoresist on any opaque or transparent substrate. As for in oil immersion configuration, in the air configuration the focus quality depends on the vertical position of the focus in the resist. The maximum structure height is defined by the working distance of

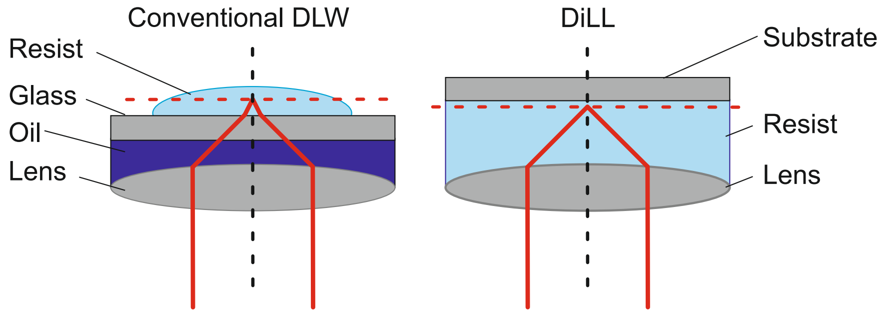


Figure 5.2.: Comparison between oil immersion and DiLL configuration

the microscope objective. See Table 5.3 for the standard hardware used for the air configuration.

Table 5.1.: Standard oil immersion configurations

objective	immersion medium	resists	substrate
63x NA 1.4	oil	IP-L 780, IP-G 780	borosilicate coverslip
100x NA 1.4	oil	IP-L 780, IP-G 780	borosilicate coverslip

Table 5.2.: Standard DiLL configurations

objective	immersion medium	resist	substrate(s)
25x NA 0.8	resist	IP-S	ITO coated glass, silicon
63x NA 1.4	resist	IP-Dip	fused silica
100x NA 1.3	resist	IP-Dip	fused silica

Table 5.3.: Standard air configurations

objective	immersion medium	resist	substrate
20x NA 0.5	air	AZ resist	glass, silicon

5.3. Sample preparation

This section covers the sample preparation processes for the following resists:

- **IP-L 780** is a liquid negative-tone photoresist formulation designed for high resolution printing in oil immersion configuration. IP-L 780 combines high resolution with low shrinkage and a high stability.

- **IP-Dip** is a liquid negative-tone photoresist formulation designed for high resolution printing in DiLL configuration only. IP-Dip features high resolution on various type of substrates. Due to its compatibility with the DiLL configuration IP-Dip structures may largely exceed the vertical piezo range as well as the working distance of the objective lens.

- **IP-S** is a liquid, but highly viscous negative-tone photoresist formulation designed for mesoscale printing in DiLL configuration only. IP-S combines low shrinkage and high stability with smooth structure surfaces. Due to its compatibility with the DiLL configuration IP-S structures may largely exceed the vertical piezo range as well as the working distance of the microscope objective.

For other resists please refer to the respective data sheets and manuals of the resist suppliers.

5.3.1. Substrates

Make sure to use clean substrates suitable for your desired configuration. See Tables 5.1 to 5.3 for the correct substrate choice in the standard configurations. The cleaning process usually consists of the following steps:

1. rinse the substrate surfaces with acetone
2. rinse the substrate surfaces with isopropanol
3. rinse the substrate surfaces with distilled water
4. blow-dry the substrate with nitrogen to reduce moisture

Using this cleaning method a droplet of resist will not spread strongly over the substrate surface. Alternative cleaning methods (e.g. oxygen plasma) might change the wetting properties of the substrate surface so that the droplet will cover a bigger area, which may also be desirable in some cases.

5.3.2. IP-L 780 sample preparation

- *Sample fixation:* Substrate(s) should be fixed properly on the appropriate sample holder using tape or Fixogum. Note that each sample position is made for a **special size and thickness** of substrates.
- *Resist application:* Apply the resist on the top side and center of the substrate. It is recommended to use a conventional pipette or spatula. Please note that spin coating of the resist is in general not necessary since merely the voxel volume will be exposed. The structure height is thus not defined by the resist thickness but rather by the height of the exposed volume!
- *Prebake:* no prebake

5.3.3. IP-Dip sample preparation

- *Sample fixation:* Substrate(s) should be fixed properly on the appropriate sample holder using tape or Fixogum.
- *Resist application:* Apply the resist on the bottom side and center of the substrate (pointing towards the microscope objective). It is recommended to use a conventional pipette or spatula. Please note that spin coating of the resist is not possible in DiLL-configuration.
- *Prebake:* no prebake

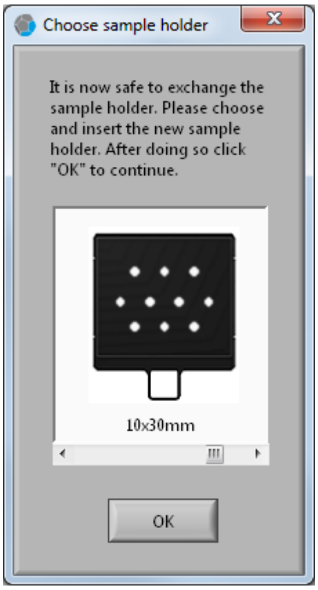


Figure 5.3.: Exchange Holder dialog

5.3.4. IP-S sample preparation

- *Sample fixation:* In case you are using an ITO coated substrate from Nanoscribe, please make sure to fix the substrate with the ITO coated side facing downwards (towards the microscope objective). We recommend the use of an ohmmeter to check which side is ITO covered. Since ITO is conductive the ITO-side has a rather low resistance of a few Ohm. Substrate(s) should be fixed properly on the appropriate sample holder using tape or Fixogum.

- *Resist application:* Apply the resist on the (ITO-coated) bottom side and center of the substrate (pointing towards the microscope objective). It is recommended to use a conventional pipette or spatula. Please note that spin coating of the resist is not possible in DiLL-configuration.

- *Prebake:* no prebake

5.4. Loading sample holder

For loading a new sample holder click **Exchange Holder** and wait for the exchange dialog to appear (see 5.3). The objective will move to the lower z-drive limit and the piezo servos will be switched off as long as this dialog is active.

Warning!
A message window pops up (see Figure 5.3) when the system is ready for the sample holder exchange. During the holder loading this message has to be active. Do not confirm it prior to finishing the loading procedure. Otherwise you have to re-click **Exchange Holder** before continuing.

Once the sample holder is correctly loaded, choose your sample holder in the dialog, then select the desired sample position within the holder and confirm the message by

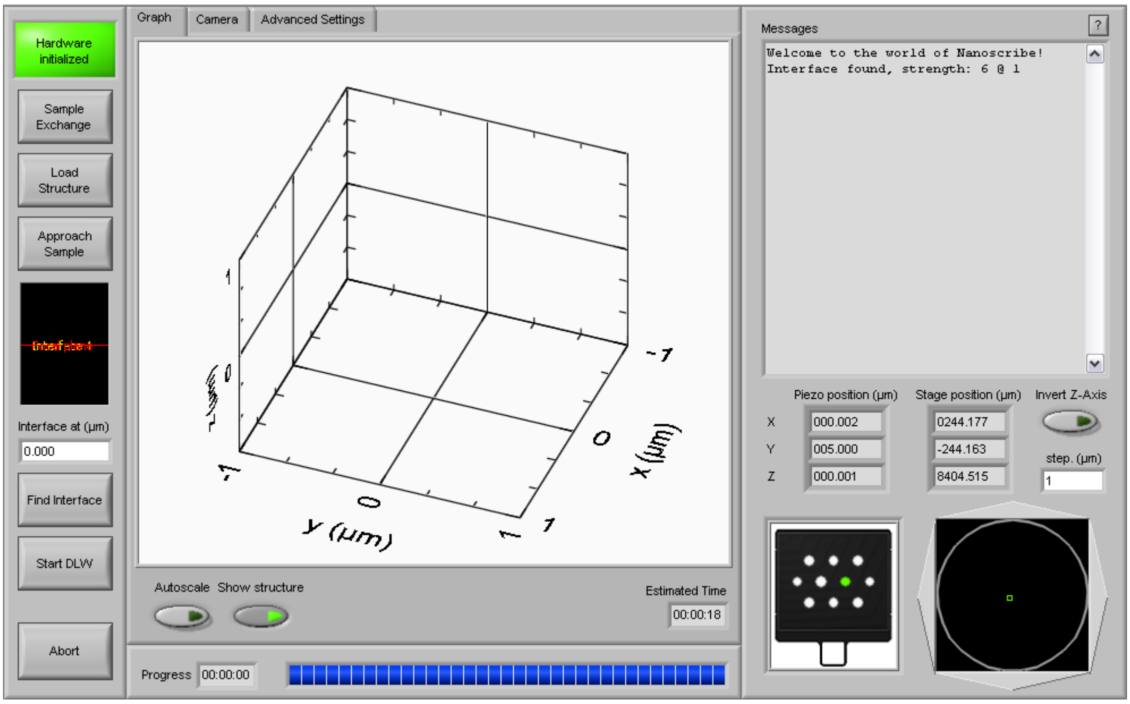


Figure 5.4.: Interface is displayed after approach

clicking on click **OK**. The system is now operational again. This is also a very good point in time to select the correct objective type.

5.5. Sample approach

The sample approach procedure is automated. You only have to choose your desired sample position on the sample holder icon via a double click. The system will address this sample position (in x-y-direction per stage movement). Once the sample position has been changed press click **Approach Sample** so that the system drives the objective up towards the substrate (in z-direction) until it finds an interface, i.e., a sufficient abrupt change of refractive index. The system is ready when the interface is displayed in the **interface icon** as shown in figure 5.4.

5.6. Loading a job

To load a GWL-file click click **Load Job** button. Choose your GWL-file and confirm your choice. When loading is done the **MessageLog** displays *yourgwlfile-job.gwl loaded*. If click **Load Job** is activated and displayed as click **Loading** NanoWrite automatically starts a pre-parsing of the loaded data at that point. If changes have been made to the file without DeScribe or the GWLc folder was not copied, NanoWrite will compile the GWL-files to create GWLc files.

Note that STL files cannot be directly loaded into NanoWrite. Please use the DeScribe import feature to create suitable GWLb files instead. Refer to Chapter 8 for details about the STL import.

5.7. Writing process

Once all prior steps are performed the writing process can be started by clicking on click **Start Job**. For many resist materials and printing configurations the printing process can be observed in the live cam. If this is not the case you can still see the process on the piezo and stage position indicators. You may also use the command `MessageOut` in your GWL files to display messages on the MessageLog during the printing process.

5.8. Unloading a sample holder

For unloading a sample holder click click **Exchange Holder** button. The microscope objective goes to the lower z-drive limit and the piezo servos are switched off.

Warning!
A dialog is displayed (see Figure 5.3) when system is ready for the exchange. During the holder unloading procedure this message has to be active. Do not confirm it prior to finishing the unloading procedure.

Carefully pull on the holder handle until the sample holder is completely detached from the system. Next either insert another sample holder or directly confirm the dialog Figure 5.3 by clicking on click **OK**.

5.9. Sample development

The development of IP-L 780, IP-Dip and IP-S is almost identical: Use a bath of PGMEA in a 25 ml beaker. To fix the substrate vertically in the beaker use the supplied substrate holder. The developing time is 20 min for IP-L 780, IP-Dip and 30 min for IP-S. Afterwards carefully pull out the substrate holder from the PGMEA bath and put it in a second beaker with iso-propanol or about 2 min. Finally blow dry the substrate very gently with nitrogen. Note that the development time is not very critical. In general you may keep your structures in the developer for a much longer time without changing the structure properties significantly. Just make sure to develop long enough to get rid of all monomeric material.

A post-curing process is recommended for some applications (Chapter 6.1.3).

5.10. Workflow overview

The basic standard operation procedure is summarized below:

Starting NanoWrite

- Start NanoWrite.
- Wait for the stage calibration warning message Figure 5.1.
- Make sure no obstacles are present for the stage movement.
- Check whether the microscope objective is at the lower z-drive position.

- Confirm the warning message by clicking on **Calibrate**.
- Wait for the initialization to finish.
- After the Exchange Holder dialog popped up, carefully insert your sample holder.
- Choose your sample holder via the slider in dialog box Figure 5.3.
- Confirm your choice by clicking on **OK**.

Choice of the sample position

- If you want to change the sample position do so by double clicking on the respective position on the **sample holder indicator**.
- Click **Approach Sample** move the objective to the interface. After a successful approach, the interface should be displayed in the **interface indicator**.
- Click **Find Interface** to test the autofocus. The MessageLog should display *Interface found* and a signal amplitude.

Loading of a GWL-file

- Click **Load Job** and choose your GWL-file.
- Wait for the loading to finish (indicated by *filename.gwl loaded*).

Starting the exposure

- Click **Start Job** to start the writing process.
- Wait until the job is finished or press **Abort** to stop the exposure.

Unloading the sample

- Click **Exchange Holder**.
- Wait for the Exchange Holder dialog to pop up, see Figure 5.3.
- Carefully pull out the sample holder.
- Confirm the dialog by clicking on **OK**.
- Proceed with the sample development.

Cleaning the objective

- If an immersion objective was used and a change of resits is planned or no subsequent job lines up, it is advised to clean the objective (Chapter A.1.1).

6. Application examples

The Photonic Professional (GT) is a very versatile tool for fabricating structures from the nanometer up to the millimeter scale. In order to give you an overview of the diverse possibilities we will introduce the following processes in this chapter:

- Mesoscale 3D printing (GT)
- High resolution 3D printing (GT)
- Maskless 2D lithography (GT)
- Real 3D trajectories (Piezo)

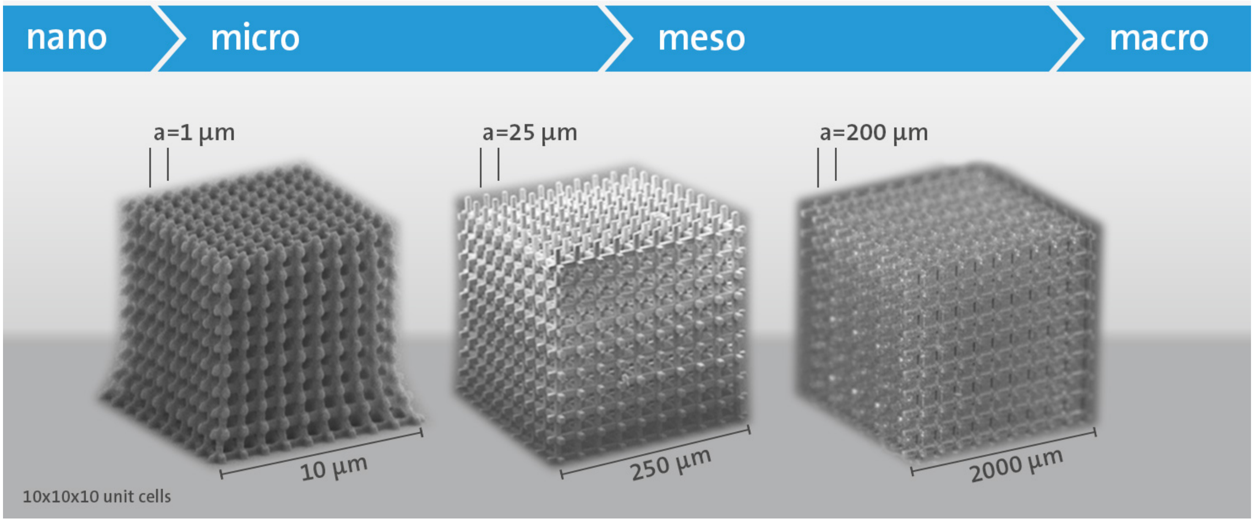


Figure 6.1.: From the micrometer to the millimeter scale: These lattice cubes were all printed using the Photonic Professional (GT) to demonstrate the versatility of the micro production process for different scales.

Each process is explained and discussed using one exemplary application or structure. This chapter is meant to guide you along these standard processes and workflows giving you the ability to quickly choose the matching hardware configuration for your desired application. Please note the tabular representation of the required hardware configuration for each process at the beginning of the sections.

Through its compact and graphic form, this chapter will also give you easy access to the main working and design principles necessary for a well printed structure. For more detailed and technical explanations please refer to the chapters 7 and 8, whereas the necessary prerequisites can be found in chapter 5.

Table 6.1.: Summary of the experimental condition

Item	Designation	Description
Objective	25x NA0.8	Immersion objective 380μm Working distance
Substrate	ITO coated Glass	Thickness 700μm
Photoresist	IP-S	Refractive index : 1.48 at 780 nm

6.1. Mesoscale 3D printing with the 25x NA0.8 objective

6.1.1. What is mesoscale printing?

Combining the highest resolution below the micrometer range with the aim of fabricating structures with dimensions above the millimeter has huge implications on the fabrication process. In this chapter, we will present the concepts and workflow for enabling you to fabricate a structure according to the following goals:

- The structure with a 3D design has geometrical dimensions exceeding the millimeter range
- It contains feature sizes down to 1 µm
- The writing time should be reasonable: the fabrication should be performed overnight

6.1.2. Materials and processes

In the example that we will develop in this chapter, and in general for targeting mesoscale applications, a general set of technical conditions should be fulfilled. The standard process concerning mesoscale printing comprises the 25x magnification objective with a writing field diameter in galvo scan mode of 400, μm, and yet with a feature sizes in the sub micrometer range, given the numerical aperture of 0.8. We furthermore use IP-S as indexed matched photoresist for printing in the DiLL configuration. Please use either ITO-covered glass substrates as we do in this example or appropriate opaque substrates in order to be able to use the interface finder.

Please refer to chapter 5.3 for an introduction on how to prepare the substrate on the sample holder.

6.1.3. Fabricating large structures

Description of the design

We now focus on the design of the structure we would like to fabricate. We want to fabricate the figure of a dancing ice skater (Fig. 8.2). In a first step, the lateral dimensions of the structure will be limited to the writing field of the microscope objective. The height is approximately 1 mm. This design contains high resolution features on the clothes of the ice skater as well as large volumes such as the body.

When opening the corresponding STL file in DeScribe, the design can be previewed as in figure 8.2.

Shell Writing

In this section we will use the STL import function of DeScribe and find a standard parameter set for a mesoscale structure approaching or reaching the millimeter scale. When importing a STL model in Describe you define the printing process by setting parameters for slicing, hatching, as well as for the structure's shell and scaffold. For a detailed description on the functions and mechanisms of Describe please refer to Chapter 8.

Each layer is then subdivided into a set of trajectories, being straight lines or concentric perimeters, that are separated from each other by the so-called **hatching distance**. Since all trajectories in a XY-plane are addressed by the galvo scanner, the writing time depends directly on the total length of the lines.

In this section we apply some strategies for reducing the writing time, such as the **shell & scaffold writing**. This writing mode consists of fabricating a dense shell delimiting the structure as well as the interior of the structure which is only partly polymerized in the form of a scaffold, helping to mechanically stabilize the structure.

Once the fabrication process is completed only a fraction of the structure will be polymerized. The topology of the structure is preserved since the shell prevents the developer to creep inside. In a second step, the remaining resist is polymerized through post processing either by heating or UV flood exposure of the sample.

When exceeding the geometrical limits of the galvo writing field diameter (400 µm with 25x NA 0.8), or of the piezo for the z-coordinates (300 µm), the design must be split into a series of blocks. All blocks are finally stitched together using the stage and the z-drive of the objective.

Once the model has been resized to 1 mm height within Describe (Fig. 6.2), we move on to the slicing menu by clicking on the button click **Slice** at the top right corner. For our example we will set the slicing distance to 1 µm as a standard and fixed value, for our combination of objective-photoresist-substrate.

By clicking on click **Fill**, we move on to the shell and scaffold parameter settings. In a first step we choose the Shell & Scaffold fill mode, and then define the main shell parameters:

- **Hatching distance** This setting defines the lateral distance in µm of two adjacent lines within a layer. This may be the lines of the shell itself (contour >1), or for the filling of the base and ceiling layers.

- **Shell contour count** should not be chosen lower than 12 for IP-S resist. For other material & objective combinations values down to 1 may be useful, too. Its purpose is to reduce deformation due to polymerization shrinkage and to prevent the developer from creeping into the structure during the development process.

- **Base slice count** defines the future interface to the substrate as the bottom part of the structure. As a standard it can be set to 5. The base layer strengthens the first lines written onto the substrate as they tend to undulate if then do not adhere strongly enough to it. This way, resist leakage during development can also be limited. Another important option for the base layer is to use a different `LaserPower` for these layers, e.g. a lower power in order to reduce the probability of bubbling in proximity to a highly reflective substrate.

Note that `LaserPower` as well as `ScanSpeed` can be chosen individually for shell, scaffold and base. They will be available as variables after the import process.

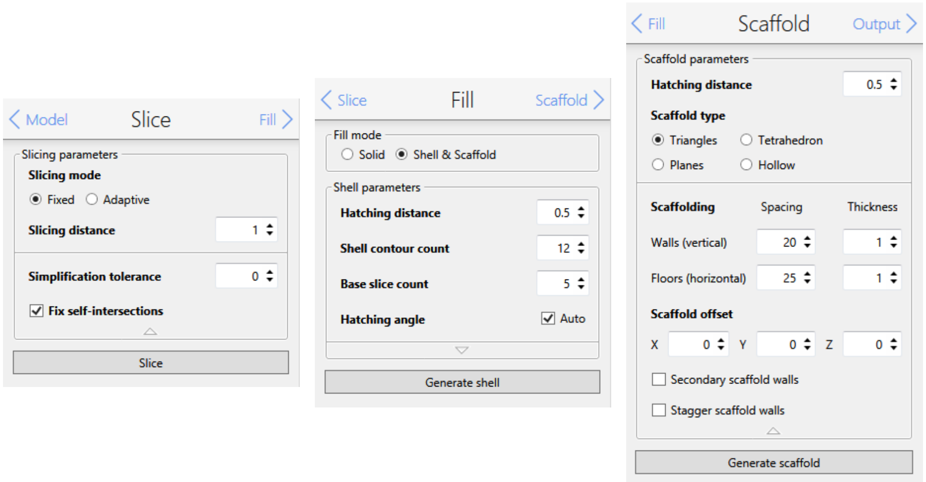


Figure 6.2.: For the 25x objective and IP-S, the default slicing distance is 1 µm and hatching distance 0.5 µm.

Table 6.2.: Table summarizing the standard parameters chosen for converting a STL file into a GWL format in the experimental conditions details in this chapter.

Step	Parameter	Value for IP-S
Slicing	Slicing Distance Hatching Distance	1 µm 0.5 µm
Shell	Shell Thickness Base thickness	12 lines 5 lines
Scaffold	Type Walls Spacing Floors Spacing Walls Thickness Floors Thickness	Triangles 20 µm 25 µm 1 line 1 line
Splitting	Lateral block size Vertical block size Angle Overlap	300 µm 150 µm 15 ° 2 µm

Once the shell parameters are set, we move on to define the scaffold parameters by clicking on the corresponding button on the top right hand corner. The default scaffold is *triangles*: a series of triangular cells separated by horizontal layers. The hatching distance maybe chosen differently than the hatching distance for the shell, but in general it works best with 0.5 µm as well. The triangle unit cells are set to 20 µm, with a thickness of 1 µm meaning the walls are made of single lines. For the vertical spacing, a similar order of magnitude is chosen, a spacing of 25 µm and single layer thickness.

The 20 µm cell size is relatively small in order to provide enough mechanical support to the shell. The single-line walls typically yield enough support. The floor planes also contribute to the stability of the scaffold and the flatness of shell walls.

The output menu allows the user to define the relevant parameters for generating an actual GWL-file. The scan mode and the hatching mode can be chosen as bidirectional for higher speed. The user may also split the structure and/or repeat the pattern in the form of an array modifier. Splitting and stitching are treated in detail in the next section. Two bullet points should be highlighted. Splitting the structure laterally is only necessary if its lateral dimensions exceed the writing field, i.e. 400 µm. This case will be treated in section 6.1.4.

If the structure height exceeds the travel range along the z-axis of the Piezo, i.e. 300 µm, the structure has to be split and the microscope z-drive is used to stitch the blocks together. In combination with angled stitching, a typical setting of the blocks' z-height is 150 µm. In that way, the individual blocks are fabricated subsequently with Piezo movements in between the layers. When each block limit is reached the z-drive of the objective is lowered by 150 µm and the Piezo height is reinitialized to its start position.

Alternatively, if the high precision of the Piezo is not needed, the z-drive can be used exclusively for the z increments.

All the parameters, that we are using in this example are summarized in the table 6.2 After saving the structure, a complete file structure is generated with respect to the given base name:

- The *basename_job.gwl* is the main GWL file meant to be loaded into NanoWrite (with a version higher or equal to 1.8). In that file the user can define the writing parameters with the GWL script commands such as the `LaserPower` or `ScanSpeed`. A standard set of parameters is automatically proposed.
- The *basename_data.gwl* includes all the data files and a summary of the STL conversion parameters. If the splitting was used, all blocks are assembled together with stage movements.
- the *basename_files* directory contains compiled GWL files, with the *.gwlb extension. These files contain the data for each single block in a binary form. They cannot be modified by the user.

Process parameters

Opening in DeScribe the *IceSkater_job.gwl* file shows us the standard header with parameters for setting some offsets and basically the two main process parameters when writing with the 25x magnification objective in IP-S that are the `LaserPower` and the `ScanSpeed`. For our example we will set the `LaserPower` to 70% and the `ScanSpeed` to 50 mm/s. The ideal setting of these parameters may vary on your system.

It is thus recommended to characterize your photoresist and your system by fabricating a series of benchmark structures varying both the `LaserPower` and `ScanSpeed`. The degree of cross-linking in IP resists is correlated with the laser intensity or energy deposited within the voxel. The range between the polymerization threshold and the threshold for bubbling is called the polymerization range, for a given photoresist-objective combination. It is a function of the laser power and the scan speed, as well as slicing and hatching distances. Aggregates (bubbles) of polymerized resist appear when the threshold for bubbling is reached.

```
% Field Parameters
XOffset 0
YOffset 0
ZOffset 0

```

```
PowerScaling 1.0
LaserPower 70
ScanSpeed 50000

FindInterfaceAt 1
% Include slicer output
include IceSkater_Array_data.gwl
```

Listing 6.1: GWL job file header with standard parameters.

We are now ready to open the *IceSkater_Array_job.gwl* in NanoWrite since the objective is already approached to the substrate. In general, it is recommended to run long term processes overnight in order to minimize the risk for having external perturbations such as vibrations induced by the door opening in the lab room.

Post Processing

The post-processing phase is divided in two steps. At first, the structure has to be developed so that the monomeric polymer is removed. Nanoscribe GmbH recommends using either PGMEA or mr-Dev 600 developer from the company MicroResist GmbH for 30 min (refer to Chapter 5).

During a post-curing step, the monomeric resist enclosed in the shell is polymerized, either through UV-curing or heat curing. We would like to provide advices, and help users start-up developing their process. However, we do not guarantee the exactitude of the parameters given in this chapter. The user is advised to search for optimal parameters for each new application.

1. The first option is **UV curing**. The required irradiation intensity depends on the curing depth. As primary guidance, LED sources irradiating around 400 nm wavelength could cure 1 mm high structures made out of IP-S resists with an intensity of 1.2 W/cm^2. For a curing depth of 2 mm into IP-S, 5 W/cm^2 would be necessary. Higher intensities can burn the resist and should be avoided. The exposure time typically ranges from 5 to 20 minutes, and the substrate distance is kept within few centimeters. Such sources that are now commonly used for curing epoxies are available from different manufactures, for instance Hönle (http://www.hoenle.de/en/4/) or Dymax (http://www.dymax.com).

2. The other option is **heat curing** by using a hot plate. For preserving the structure from high thermally induced stress, a ramped profile must be set-up. Heat curing occurs at about 190 ° C. The profile must aim at reaching this temperature with a slow ramp-up, and then keep the substrate at its curing temperature 190 ° C for 5 min.

Note that UV-curing is more gentle than heat curing, in that the resulting structures are less subject to stress and deformation. Heat curing is not possible with **IP-Dip** and **IP-L 780**.

6.1.4. Beyond the galvo writing range

Splitting

In the last section, the issue of the height of the ice skater was mentioned. Indeed, a 1 mm tall structure largely exceeds the capacity of the Piezo range, that is 300 µm in all three axes. Even though the galvo scan mode is used, the layer z-position is updated with a Piezo movement. Two options address this issue.

On the one hand, the Z-drive of the microscope can be used to move the objective away from the substrate. This solution is easy to implement, however the precision of the interlayer distance cannot be guaranteed especially for distances below one micrometer.

On the other hand, the structure can be split into several blocks piling up to form the structure. Within each block, the layer position is updated through Piezo movements, while the relative position of each block is updated through a movement of the z-drive by the block size. In this fashion, the error is reduced to few wide range movements so that fine features remain for the major part unaffected.

Similarly, the splitting algorithm can be applied when the lateral dimensions of the structure exceed the writing field recommended for the objective. With the 25x magnification objective, as detailed at the beginning of this chapter, the writing range should be limited to 400 µm. Otherwise, the splitting algorithm has to be used with the parameters described in Table 6.2.

The general rule explicated above applies for determining the block size with the 25x magnification objective:

1. If the structure lateral dimensions exceed 400 µm the lateral block size in X and Y directions must be set to 300 µm;

2. If the structure is taller than 300 µm the vertical block size must be set to 150 µm. It is important to notice that if the dimensions require only vertical splitting, the blocks lateral dimensions have to be set greater than that of the structure itself.

These options are available in the last menu of the STL import wizard in DeScribe.

Stitching

A split structure is then fabricated in a sequential process, in which each block is completely polymerized according to the design parameters, in galvo scan mode. The assembly of all the different blocks is managed automatically by DeScribe and implemented in the *basename_data.gwl* file. The blocks located at the same floor level are assembled with stage movements. When a level is complete, the microscope z-drive position is moved by the block height distance.

The action of assembling together a set of blocks constituting the full structure is called **stitching**.

When dealing with vertical stitching, the impact of the previously polymerized blocks on their neighbors polymerization has to be considered. This is the purpose of the angled stitching option.

Consider the fabrication of two adjacent cubes of 100 µm height, stitched together on a substrate. Once the first block is fabricated, the stage moves by the block width laterally and starts the fabrication of the second unit. Since the lateral walls of the first unit are perpendicular to the focal plane of the objective, this vertical wall induces refraction, due to the small optical index shift after polymerization. In the wall vicinity the energy distribution is flattened so that a perfect junction of the two blocks is not possible.

This shadowing effect can be reduced in two ways. One can choose a very low block height, below 25 µm so that the side walls of adjacent units are not impacting the energy distribution of the laser beam around the focal point. However, this solution results

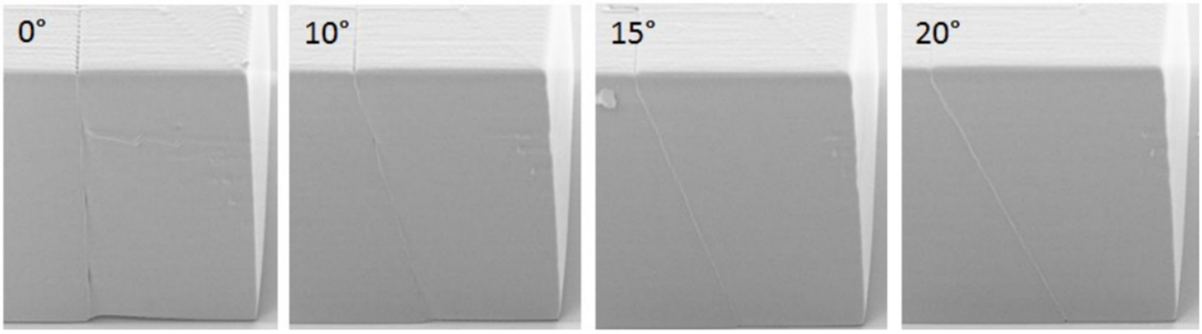


Figure 6.3.: Fabrication of a 100 µm wide cube split into two blocks stitched with various angles. The shadowing effect at the stitch interface can be observed at low angle

in a larger number of block layers and therefore a greater number of blocks. The time dedicated to stitching will be greater accordingly.

The second option, is **angled stitching**. Blocks are not anymore defined as cubes, instead cuboids with a given wall angle. Thanks to this angle, it can be arranged that even at the stitching junctions the side walls of previous blocks are not shadowing the laser beam cone as going out of the objective. The angle of the stitching blocks is determined as an option in DeScribe and must be set to at least 15°. An example is represented on Figure 6.3.

The block height of 150 µm as mentioned in the previous paragraph is applicable only when the angled stitching is enabled.

Besides, setting a **stitching overlap** can be set for reducing the risk of failed stitching junctions due to discrepancies in stage movements. This overlap extends the block dimension at walls in contact with another block, so that it can be ensured that the junction is fully polymerized. An overlap of 2 µm is recommended.

Once again, the user should refer to the Table 6.2 in which the standard parameters, used for converting a STL file into GWL, are detailed. The numerical values are only valid in the experimental conditions explicated at the beginning of the chapter: the combination of the objective, photoresist and substrate.

6.2. High resolution 3D printing with the 63xNA1.4 obj.

6.2.1. Scope

This chapter addresses relevant steps allowing printing of almost arbitrary formed structures with sub-micron resolution. It contains general information about the appropriate hardware and materials, an exemplary preparation of a typical structure and the standard printing parameters.

6.2.2. Hardware and resists

Objective

The objective choice is decisive for the kind of results you are aiming at. The best resolution with the system is achieved when using the 63x magnification objective, with a high numerical aperture (NA=1.4). On this account it is the standard objective for all kind of applications requiring high resolution, e.g. photonic crystals and meta-materials.

The 63x objective
This objective can be combined with various resist and substrates and can be utilized in the Oil Immersion- as well as in the DiLL-configuration. Please refer to chapter 5 for further information on this issue since an incorrect combination may have undesirable consequences.

Resists

To enable the maximum performance of the PPGT in high-resolution printing it is recommended to use the tailored IP-Dip resist. IP-Dip may only be used in the DiLL-configurations described in section 5. The usage of IP-Dip allows a printing process with minimized constant spherical aberrations and without limitations in structure height.

Substrates

The autofocus system only works for resists having a slightly different refractive index than the substrate, as explained in chapter 7.5. In order to give many possibilities to the user, the DiLL-holder supports different kinds of substrates. Please make sure to have a sufficient refractive index difference $\Delta n = n_{substrate} - n_{resist}$. For the 63x NA 1.4 objective the index difference should be $\Delta n > 0.05$ When printing in IP-Dip the usage of the standard DiLL-substrates is recommended. Please refer to chapter 5.3 for details on the sample preparation process. Once the sample is prepared and fixed on a DiLL sample holder it can be loaded into the system according to the standard procedure described in Chapter 5.4.

Having approached the substrate with the objective, that is to say, having the objective touching IP-Dip resist the writing process can be started, so that the top of the structure is reached by getting away from the interface.

Table 6.3.: Required components for standard high-resolution printing applications

Item	**Designation**	**Description**
Objective	63x NA1.4	Immersion/DiLL obj. working distance: (170+190) µm
Resist	IP-Dip	refr. index: 1.52 at 780 nm solvent free
Substrate	DiLL-Substrate (fused silica)	Dimensions: $(25 \times 25 \times 0.7)$, mm^3 refr. index: 1.46 at 780 nm

6.2.3. Printing high-resolution structures

Design

The structure chosen for the exemplary work flow is a woodpile as shown in the DeScribe preview in Figure 6.4. Please refer to Chapter 8 for further information on the programming basics in DeScribe. The woodpile dimensions are $(3\times10\times10)$ µm^3 and the lateral rod spacing is 0.7 µm. The code data of the woodpile can be found in the *woodpile_data.gwl* from the GWL-examples folder.

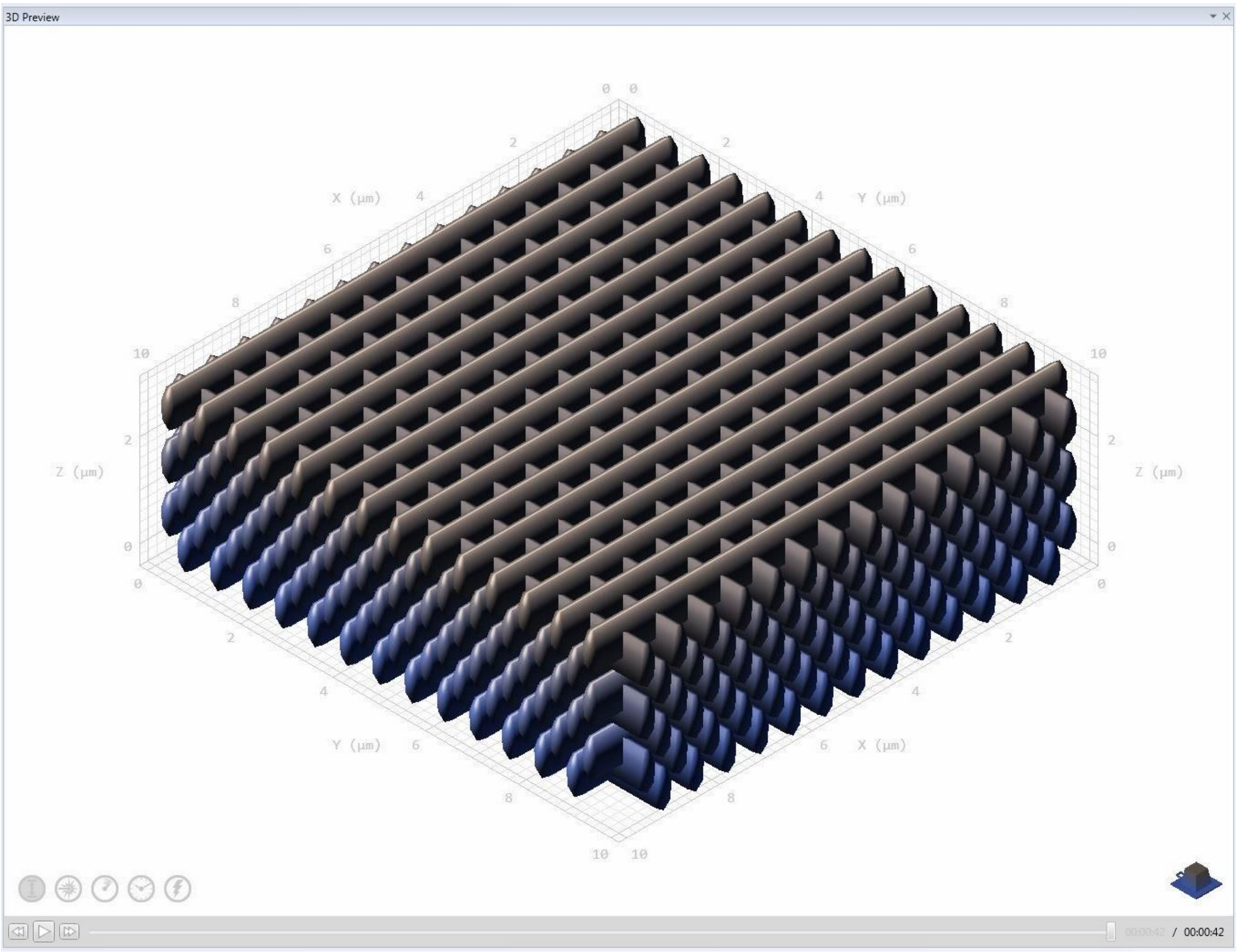


Figure 6.4.: DeScribe preview of the programmed woodpile that is used for the exemplary high-resolution printing work flow. The preview rendering mode is set to billboard lines striving a more realistic picture of the polymerized lines.

Process parameters

The general header containing the process parameters is usually located in a job file. This job file furthermore includes one or more data files in which the coordinates are saved. The standard process parameters that are also used in the *woodpile_job.gwl* can be found in Listing 6.2.

```
% System Initialization
InvertZAxis 1

% Printing configuration
GalvoScanMode
ContinuousMode

% Printing parameters
PowerScaling 1.0
LaserPower 50
ScanSpeed 10000

FindInterfaceAt 0.5

include woodpile_data.gwl % including the structure code data
```

Listing 6.2: Exemplary job file *woodpile_job.gwl* containing all relevant process parameters for high-resolution printing in DiLL-configuration

The commands and parameters from the *woodpile_job.gwl* are good to start with for your own structures. In most cases you will have to tune the `LaserPower` and adapt the `FindInterfaceAt` value to match the requirements of your structure. The commend `InvertZAxis` should always be set to 1 µm when printing in the DiLL-configuration. Please refer to chapter B for further information on all available commands in DeScribe. Once all prior steps are performed the printing process can be started by clicking on the **Start Job** button. The printing process can be observed in the live camera window.

Note!
It is also possible to write high-resolution structures in the immersion-configuration when IP-Dip does not come into question for whatever reasons. In this case one should consider spherical aberrations and set `InvertZAxis` to 0 µm. For further information please refer to chapter 5.

6.3. Maskless 2D lithography with 20x NA0.5 obj.

In this chapter we will discuss the processes for writing structures with the 20x NA 0.5 air-objective in positive tone photoresists with the GT option. We will focus on the processing parameters of four different types of AZ-resists:

- **AZ 5214E** is a thin film resist used for low resolution high speed patterning (section 6.3.1).
- **AZ 9260** is a thick film resist used for high aspect ratio patterning (section 6.3.2).
- **AZ 40XT** is a thick film resist used for low aspect ratio patterning (section 6.3.3).
- **AZ MIR 701** is a thin film resist used for high resolution, high speed patterning (section 6.3.4).

Please note that AZ-resists are very sensitive to the environmental conditions. The ideal process parameters may thus vary in your laboratory and you probably will have to adapt most of them to your conditions. For the recipes within this chapter we used a *Laurell 650SZ 6NPP/LITE spincoater* for spinning of adhesion promoter and resists. Furthermore we put our substrates in direct contact with the hotplate during all baking steps.
The photoresist layer thicknesses discussed in this chapter are smaller than the vertical resolution of the interface finder in combination with the 20x NA0.5 air-objective. The software thus does not differ between the air-resist and the resist-substrate interfaces. Instead in most cases the interface finder will position the laser focus somewhere in between these two interfaces. For writing with the 20x NA0.5 air-objective we thus firstly need to seek the ideal vertical working position, where the laser focus is in the desired writing layer. Try different combinations of `FindInterfaceAt` and `ZOffset` to find your ideal setting.

6.3.1. AZ 5214E on silicon

Substrate Preparation

first step	Bring resist and adhesion promoter to room temperature (small amount is sufficient)	
cleaning	wipe and rinse with acetone, isopropanol and distilled water in few fast iterations to avoid staining. Blow-dry it afterward.	
desorption	substrate on hotplate at 120 °C for 10 min	
adhesion promoter	TI PRIME (to apply immediately after desorption at ambient temperature 21°C). Cover substrate completely.	
	coating	Ramp +: 300 rpm/s for 10 s
		Speed: 3000 rpm for 20 s
		Ramp -: 300 rpm/s for 10 s
	hotplate	2 min at 120 °C (coating activation, immediately after coating to avoid re-adsorption)

Resist Coating

delay	wait a few minutes to cool down substrate
coating	resist temperature: ambient (21 °C)
	Ramp +: 1500 rpm/s for 5 s
	Speed: 6000 rpm for 30 s → 1.23 µm
	Ramp -: 500 rpm/s for 12 s

Softbake and Rehydration

delay	none
hotplate	110 °C for 50 s (accurate)
rehydration	few minutes

Exposure

delay	use substrates the day of preparation	
writing	scanspeed	8000 µm/s
	number of layers	1

Development

PEB	none	
development	developer	AZ 726 MF in beaker
	concentration	undiluted
	temperature	ambient (21 °C)
	time	90-100 s optimum (80 s/ µm)
	stopper	distilled water for 10 s

6.3.2. AZ 9260 on ITO-coated glass

Substrate Preparation

first step	Bring resist and adhesion promoter to room temperature (small amount is sufficient)	
cleaning	wipe and rinse with acetone, isopropanol and distilled water in few fast iterations to avoid staining. Blow-dry it afterward.	
desorption	substrate on hotplate at 120 °C for 10 min	
adhesion promoter	TI PRIME (to apply immediately after desorption at ambient temperature 21°C). Cover substrate completely.	
	coating	Ramp +: 300 rpm/s for 10 s
		Speed: 3000 rpm for 20 s
		Ramp -: 300 rpm/s for 10 s
	hotplate	2 min at 120 °C (coating activation, immediately after coating to avoid re-adsorption)

Resist Coating

delay	wait a few minutes to cool down substrate
coating	resist temperature: ambient (21 °C)
	Ramp +: 1200 rpm/s for 1 s (1000 rpm/s for 2 s)
	Speed: 1200 rpm for 30 s → 13 µm (2000 rpm for 30 s → 9 µm)
	Ramp -: 400 rpm/s for 3 s (400 rpm/s for 5 s)

Softbake and Rehydration

delay	wait for 5 min for partial solvent evaporation
	110 °C for 6.5 min
rehydration	30 min at 50% relative humidity

Exposure

delay	use substrates within 48h	
writing	hatching distance	1.2 µm
	scanspeed	3500 µm/s
	number of layers	2 for 13 µm resist layer thickness
	layer distance	5 µm

Development

PEB	none	
development	developer	AZ 400K in beaker
	concentration	1:4 (developer:distilled water)
	temperature	ambient (21 °C)
	time	8.5 min (40 s/ µm)
	stopper	distilled water for 10 s

6.3.3. AZ 40XT on silicon

Substrate Preparation

first step	Bring resist and adhesion promoter to room temperature (small amount is sufficient)	
cleaning	wipe and rinse with acetone, isopropanol and distilled water in few fast iterations to avoid staining. Blow-dry it afterward.	
desorption	substrate on hotplate at 120 °C for 10 min	
adhesion promoter	TI PRIME (to apply immediately after desorption at ambient temperature 21°C). Cover substrate completely.	
	coating	Ramp +: 300 rpm/s for 10 s
		Speed: 3000 rpm for 20 s
		Ramp -: 300 rpm/s for 10 s
	hotplate	2 min at 120 °C (coating activation, immediately after coating to avoid re-adsorption)

Resist Coating

delay	wait a few minutes to cool down substrate
coating	resist temperature: ambient (21 °C)
	Ramp +: 1125 rpm/s for 4 s
	Speed: 4500 rpm for 40 s → 13 µm
	Ramp -: 500 rpm/s for 9 s

Softbake

delay	wait for 5 min for partial solvent evaporation
hotplate	step 1: 70 °C for 20 s
	step 2: 100 °C for 2 min (including ramp-up)
	step 3: 126 °C for 5 min (including ramp-up)

Exposure

delay	use substrates the day of preparation	
writing	scanspeed	typical 2000 µm/s, > 3000 µm/s possible
	number of layers	2
	layer distance	6 µm

Development

PEB	105 °C for 90 s	
delay	wait for 1 min to cool down substrate	
development	developer	AZ 726MIF in beaker
	concentration	undiluted
	temperature	ambient (21 °C)
	time	70 s (shorten this time in case of adhesion issues)
	stopper	distilled water for 10 s

6.3.4. AZ MIR 701 on fused silica

Substrate Preparation

first step	Bring resist and adhesion promoter to room temperature (small amount is sufficient)	
cleaning	wipe and rinse with acetone, isopropanol and distilled water in few fast iterations to avoid staining. Blow-dry it afterward.	
desorption	substrate on hotplate at 120 °C for 10 min	
adhesion promoter	TI PRIME (to apply immediately after desorption at ambient temperature 21°C). Cover substrate completely.	
	coating	Ramp +: 300 rpm/s for 10 s
		Speed: 3000 rpm for 20 s
		Ramp -: 300 rpm/s for 10 s
	hotplate	2 min at 120 °C (coating activation, immediately after coating to avoid re-adsorption)

Resist Coating

delay	wait a few minutes to cool down substrate
coating	resist temperature: ambient (21 °C)
	Ramp +: 1750 rpm/s for 2 s
	Speed: 3500 rpm for 30 s → 0.9 µm
	Ramp -: 500 rpm/s for 7 s

Softbake and Rehydration

delay	none
hotplate	90 °C for 70 s
rehydration	fast rehydration (within minutes)

Exposure

delay	use substrates the day of preparation	
writing	hatching distance	0.7 µm
	scanspeed	7000 µm/s
	number of layers	1

Development

PEB	none	
development	developer	AZ 726MIF in beaker
	concentration	undiluted
	temperature	ambient (21 °C)
	time	90 s (100 s/ µm)
	stopper	distilled water for 10 s

6.4. Real 3D trajectories - Piezo Lithography

6.4.1. Motivation

The Galvo option of the Photonic Professional (GT) offers an incredible speed up of the printing process. However, with its full 3D capabilities the Piezo still excels where arbitrary trajectories with the highest accuracy and smooth surfaces are required, for example helices or angular beams. Basically all types of structures can be produced solely with the Piezo, but writing time will often be the limiting factor, since galvo offers up to 100-times faster writing.

This chapter will introduce the most important settings for writing with the Piezo. Further details can be found in 7.8.

6.4.2. Materials and Configuration

The `PiezoScanMode` is based on the Fixed Beam Moving Sample (FBMS) principle. That is, merely the center of the objective is used and therefore all objectives may be applied for printing. Table 6.4 shows a standard setup for piezo writing in oil immersion configuration. Another typical example is depicted in table 6.5, showing how to write by Piezo in DiLL configuration. In the latter case the 100x NA1.3 objective may be replaced by the 63x NA1.4 objective, the only difference being the lower accuracy of the interface finder due to the smaller magnification.

Table 6.4.: Example of Piezo Oil Immersion writing

Item	Designation	Description
Objective	63x NA1.4 or 100x NA1.4	Immersion objective 360 / 340 µm Working distance, minus 170 µm substrate
Substrate	borosilicate glass	Thickness 170 µm
Photoresist	IP-L 780	Refr. index: 1.48 @ 780 nm

Table 6.5.: Example of Piezo DiLL writing

Item	Designation	Description
Objective	63x NA1.4 or 100x NA1.3	Immersion objective 360 µm Working distance
Substrate	DiLL substrate	Thickness 700 µm
Photoresist	IP-Dip	Refr. index: 1.52 @ 780 nm

In `PiezoScanMode` all programmed coordinates are addressed by the piezo. The maximum continuous writing range of the piezo is 300x300x300 µm^3. All programmed coordinates have to lie within this writing volume. If parts of the programmed structure exceed this volume a warning message is shown in the message log of NanoWrite (see figure 6.6). Please note that DeScribe is not aware of hardware limits.

The piezo is mounted on the motorized stage. The sample holder and with it the substrates in turn are mounted on the piezo. In this writing mode all coordinates are relative

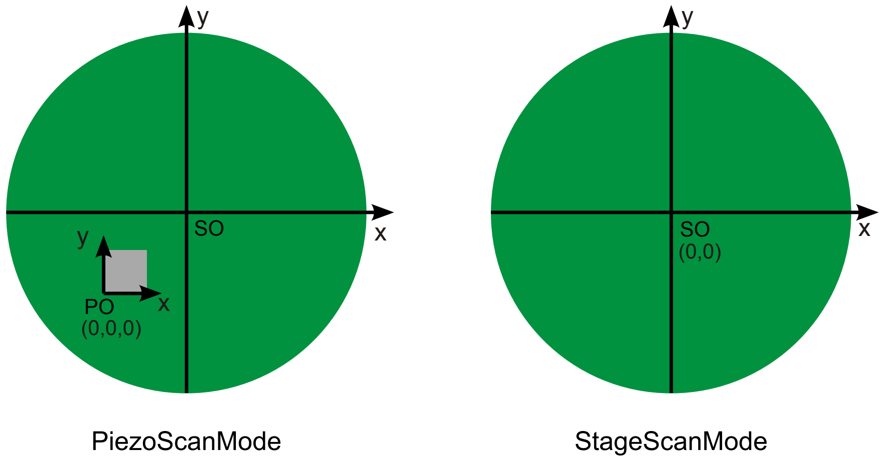


Figure 6.5.: Origin of the coordinate system depending on the ScanMode: PO=Piezo Origin; SO=Stage Origin. The gray square indicates the addressable range of the piezo.

to the current position on the substrate. Moving the substrate by a movement of the motorized stage thus allows for addressing plenty of piezo writing volumes on one substrate (see figure 6.5).

In PiezoScanMode the coordinate system moves with the stage, whereas the one for StageScanMode is fixed to the sample center (which is predefined in the `exchanger.ini`). Note that the Piezo origin (PO) is always equivalent to the current stage position.

6.4.3. Piezo PerfectShape writing

The combination of PiezoScanMode and ContinuousMode together with the fully automated path optimization PerfectShape is the most straight forward mode for piezo writing. These settings allow for writing structures consisting of continuous lines without any further knowledge of the physics of the piezo. Important is just to understand that there is always a trade-off between the best structure quality in terms of precision and the writing time.

Note!
This chapter exclusively deals with the Piezo PerfectShape algorithm and its settings. For the GalvoScanMode there also is a PerfectShape optimization, which can neither be deactivated nor adjusted. So aside from the fact that both of them optimize speed and structure quality, Piezo and Galvo PerfectShape have no common ground!

For using PerfectShape make sure that the header of your code contains all following commands:

```
PiezoScanMode
ContinuousMode
ConnectPointsOn
PerfectShapeIntermediate %choose desired profile: Fast / Intermediate /
    Quality
```

Listing 6.3: A typical PerfectShape header

How PerfectShape works

The inertia of the piezo plus sample holder and substrates makes it difficult to write structures where acceleration is necessary. Circles for example demand acceleration throughout the whole trajectory. Therefore, `PerfectShape` was developed. It dynamically adapts the scan speed and the submitted coordinates to the desired trajectory. By that, complex features are written with high precision while the total writing time is minimized.

As an analogy, when driving a car on a winded road, the speed has to be constantly adapted to stay in lane. The same applies for the piezo when writing two- or three-dimensional structures. Of course it would like to write as fast as possible, but in order to stick to the programmed path, with all curvatures and corners, a velocity adaptation is necessary. Furthermore, the localized polymerization of a resist depends on the deposited dose. So to attain constant line widths at a non-constant writing velocity, the dose has to be kept constant. That is, the laser intensity also has to be adapted to the local piezo velocity. Piezo PerfectShape integrates these two features: trajectory control and laser power adaptation.

Hardware limits when using PerfectShape

Note!
Writing with PerfectShape often requires a writing area slightly larger than the area of the programmed structure. Therefore, it is necessary to program your structures below the limits of the piezo writing volume. An easy way to avoid hitting the limits is to use offsets. Typically 5 µm in x- and y-direction are enough.

In case the piezo range is exceeded for the programmed structure, NanoWrite shows a warning in the message log (see figure 6.6). The higher the average speed of your structure, the larger is the average exceeding distance of PerfectShape. These distances are specified in the warning message for each axis.

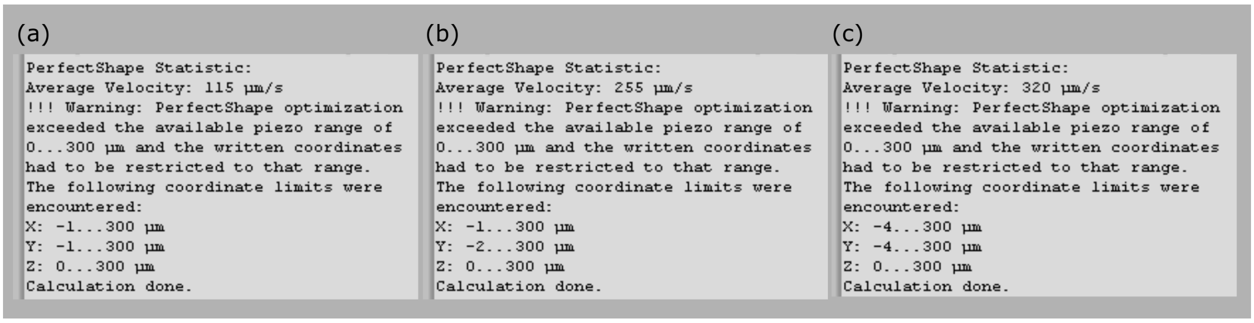


Figure 6.6.: Warning message for exceeding of piezo range due to different PerfectShape modes: (a) PerfectShapeQuality, (b) PerfectShapeIntermediate, (c) PerfectShapeFast.

To shift structures to fit into the piezo range for the respective PerfectShape mode, using offsets is recommendable. For the example of figure 6.6 (c) the adapted GWL header, could look as follows:

```
PiezoScanMode
ContinuousMode
```

Table 6.6.: The three predefined profiles of the PerfectShape algorithm. The command can also be PerfectShape followed by the argument 0 to 3.

PerfectShapeOff	With ConnectPointsOn the piezo will move at constant velocity, so low speed is necessary throughout the trajectory for achieving same quality as with PerfectShape active (see section 7.8.1 for further information).
PerfectShapeQuality	This mode ensures highest trajectory accuracy, at the expense of rather long writing time. It is used for writing three-dimensional structures with tightest radii, corners or bends.
PerfectShapeInter-mediate	This mode makes a compromise between the total writing time and the trajectory accuracy.
PerfectShapeFast	This mode minimizes the writing time by maximizing the writing speed. The trajectory is less accurate.

```
ConnectPointsOn
PerfectShapeFast
XOffset 5 %avoids exceeding of the piezorange in x
YOffset 5 %avoids exceeding of the piezorange in y
ZOffset 1 %avoids exceeding of the piezorange in z
```

Listing 6.4: Avoid exceeding of piezo range by using offsets

PerfectShape Profiles

With PerfectShape the path transmitted to the piezo is modified for taking inertia into account. Thus, maximizing scanning velocity while keeping structure quality high. But of course there is a trade-off between quality and total processing time. This is reflected by the three available PerfectShape profiles: Fast, Intermediate and Quality. The respective commands are compared in table 6.6. You have to choose the profile according to your demands.

There are two ways of placing the command. You can use either PerfectShape followed by an argument between 0 and 3, or use the corresponding alias like in the table, to apply the respective profile.

You can also manually choose the desired profile in the **PerfectShape Settings** menu shown in figure 6.7. It is accessible via the respective button in the "Advanced Settings" tab. Use the drop down box to select desired PerfectShape profile. click **LaserPower Settings** gives access to the laser power adaptation settings, which is explained in section 7.9.1. click **Calibrate Device** initiates a hardware calibration of the PerfectShape parameters for the current mass load.

PerfectShape calibration
After newly installing NanoWrite and whenever your hardware configuration changes, for example using different sample holders or different substrates, a calibration procedure is necessary. This is done by click **Calibrate Device**. Do not touch the system during calibration, as vibrations disturb the procedure and might lead to reduced writing accuracy.

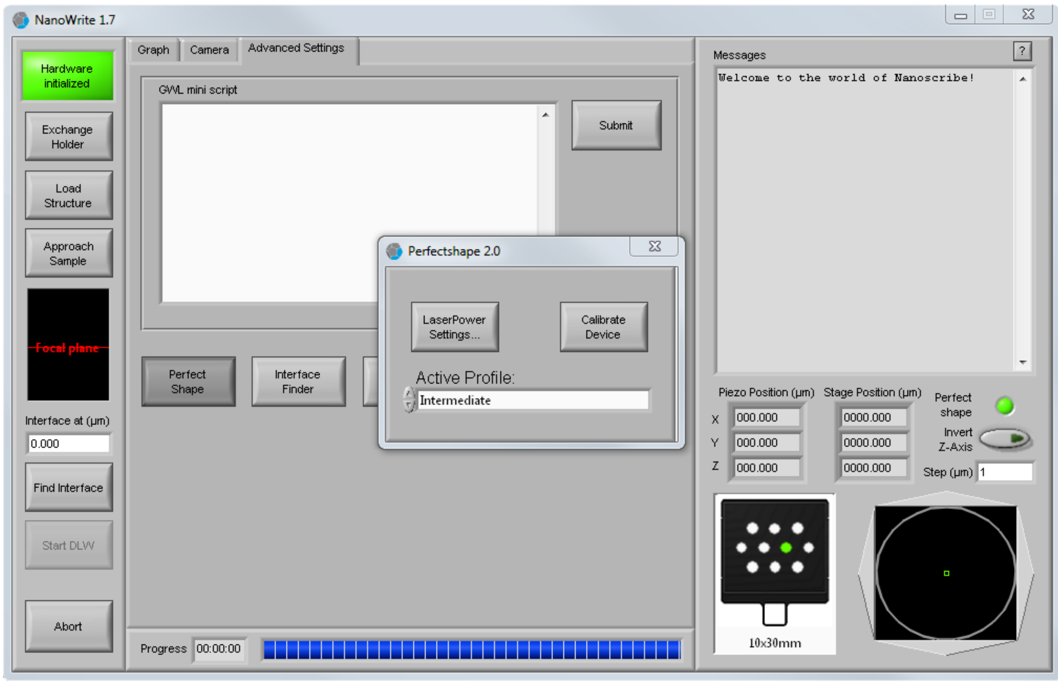


Figure 6.7.: To configure PerfectShape use click **PerfectShape Settings** in "Advanced Settings" tab.

6.4.4. Alternative operation modes in PiezoScanMode

In general, an operation mode refers to the way the laser power, velocity and trajectory are managed during writing. The resulting line parameters (lateral and axial width, resolution) depend strongly on this.

To directly drive the piezo for individual control or to create dots instead of lines additional commands need to be introduced. Two main options are available. Either writing a set of points located at the given coordinates (`PulsedMode`), or writing continuous lines passing through these points (`ContinuousMode`), like it is used for PerfectShape writing. In Continuous Mode the laser is switched on at the first point and switched off when reaching the last, whereas in Pulsed Mode the laser is continuously switched on and off (pulsed) for every point.

The piezo basically addresses a given point on the fastest route with maximum speed. To ensure a controlled speed and trajectory each line segment is automatically interpolated with intermediate points that are transferred one by one at a specific update rate to the piezos actuators. Thus, the local velocity will be product of the piezo update rate and the point distance.

This automatic interpolation is activated by the command `ConnectPointsOn`, which must be placed in the GWL-header. The counter command is `ConnectPointsOff`. It may only be used by experienced users, and only after reading section 7.8.1 carefully.

Section 7.8 explains in detail how the scanning velocity and the trajectory control of the piezo are adjustable.

Caution!
`ConnectPointsOff` might trigger very high driving frequencies that can cause serious damage to the piezo. Its resonance frequency (about 100 Hz) should never be reached. We generally recommended to stay below 10 Hz.

7. Reference GWL and NanoWrite

In chapters 5 and 6, the first steps into 3D Laser Lithography were presented, enabling you to write simple structures and to choose the adapted scanning modes and scanning parameters. In the following chapter 7.1, basic gwl programming, advanced settings, loops, stage and piezo movements and much more will be covered from a programming point of view. The aim is to gain some flexibility when building more complex and demanding scripts for any kind of design.

Not all commands will be dealt with. Wherever necessary, technical and scientific explanations will be given for a deeper understanding of a command's purpose. For a tabular listing of all commands with short definitions please refer to appendix B. All following units are given in µm and ms respectively, unless otherwise noted.

7.1. GWL-programming basics

This chapter provides the basic knowledge of programming 2D and 3D structures in GWL. For advanced programming features please refer to chapter B in general, and more specifically sections 7.3 and 7.8. Here the reader will be taught how to program structures with the most common writing mode using the piezo. These basic principles also apply using the galvo writing mode. You may use either the DeScribe editor with syntax highlighting of commands for programming GWL-files or any other text editor. You just have to make sure to use the *.gwl* file extension when saving your programmed files.

7.1.1. Writing parameters and hardware

This basic programming section focuses on the different writing parameters, writing modes and hardware choices. Therefore standard parameters are used. At the top of each following script example, the standard header as presented in appendix B.12 should always be present before loading it into NanoWrite.

The hardware setup is standard as well. All parameters in this chapter are adapted for the three standard objectives: 100x oil immersion objective with NA1.4, 100x NA1.3 and 63x NA1.4. Furthermore this chapter is based on the 10x30 mm sample holder, 30 mm round or 25 mm square cover slide substrates with 170 µm thickness and a motorized stage. We will use a droplet of Zeiss immersion oil on the bottom side of the substrate and a droplet of IP-L 780 on the top side of the substrate. Please refer to Chapter 5.3 for details on the sample preparation.

7.1.2. Programming coordinates

In order to write any structure with the laser, coordinates in the resist need to be told to the system. In the piezo standard mode we achieve this by programming line segments that are addressed subsequently by moving the sample with respect to the fixed laser focus. During these line segments the laser is running at the programmed laser power. The line segments are defined with a three column array for x-, y- and z-coordinates, given in μm. For a single straight line from $x = 0$ to $x = 10$ along the y-axis of the Piezo we thus program:

```
0       0       0
10      0       0
write
```

Listing 7.1: Line

The `write` command at the end of the coordinate array tells the system that this is the end of the current line segment. This line segment is then addressed with the laser continuously switched on at the `LaserPower` set. For a little bit more complex structure, a square with 10 µm side length, the GWL-code looks like this:

```
0       0       0
10      0       0
10      10      0
0       10      0
0       0       0
write
```

Listing 7.2: Square

So far everything had been in one z-plane. Let's have a look at a first real 3D structure.

Let's extend the square using several layers with increasing z-coordinate and on top two diagonal lines from corner to corner of the square. The corresponding GWL code of the last two layers and a preview in figure 7.1.

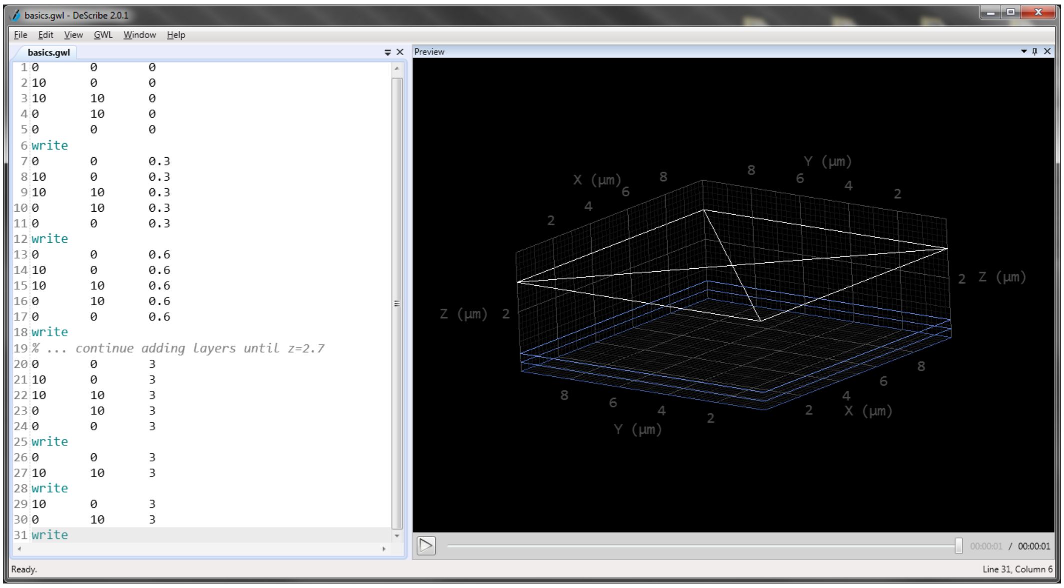


Figure 7.1.: View on DeScribe of the first 3D example.

So far, it was explained how to program very simple 2D and 3D structures using GWL-programming. The next section will give further insights about GWL-programming and some basic commands and concepts that make programming faster and easier.

Decimal Separators

Make sure to use points as decimal separators. Commas cannot be read by the Photonic Professional (GT) since the system operates with US regional settings.

7.1.3. Simple loops

In this section we will improve the programming of the previous 3D structure and make the code shorter. For this purpose we will use the GWL-commands `Include`, `ZOffset`, `Repeat` and `AddZOffset`. Before using these commands we have to save the square programmed in 7.1.2 as GWL-file in a folder of your choice. Let's name this structure *square1.gwl*. We now open a new GWL-file which we name *include.gwl* and save it in the same folder as square1.gwl. First of all we paste the content of the header defined in section 7.1.1 into this new GWL-file. We can now add the previously saved *square1.gwl* from the current GWL-file via the `Include` command as follows:

```
Include square1.gwl
```

Listing 7.3: include

This command includes all lines of the specified GWL-file to the current GWL-file. To build the 3D structure which we programmed in section 7.1.2 we have to include *square1.gwl* 10 times and adjust the z coordinate for each layer accordingly. This is done with the

ZOffset command. ZOffset adds the value in μm to all given z-coordinates addressed in the code after the command (i.e., until another ZOffset has been commanded). XOffset and YOffset work respectively for the x- and y-coordinates. You can see the corresponding code and the preview in figure 7.2.

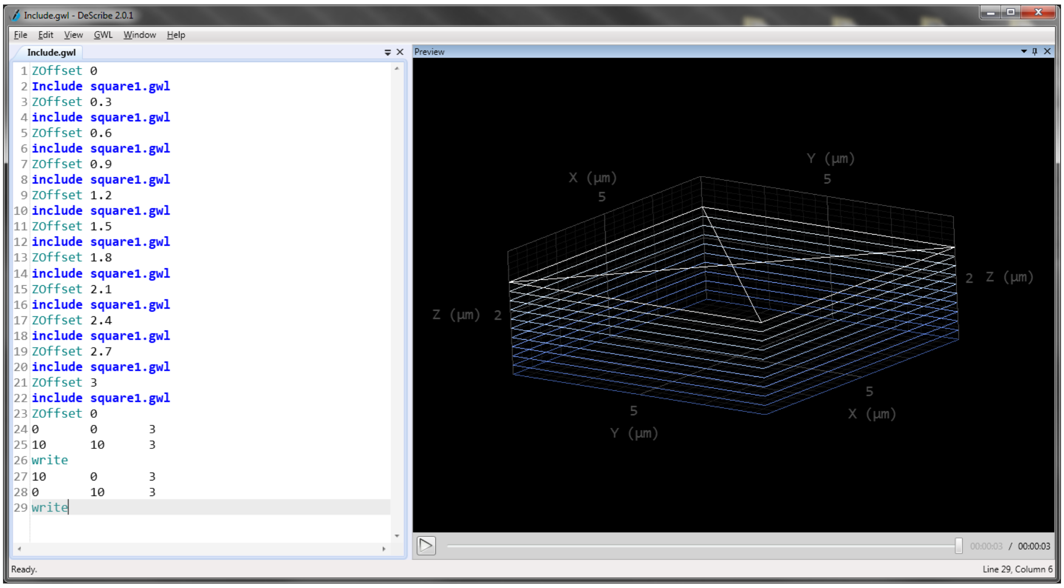


Figure 7.2.: DeScribe screenshot of the improved 3D example, with the Include command.

Note that the ZOffset is set back to zero before adding the two diagonal lines on top of the structure, since this layer is directly programmed at z=3.0.

To make the GWL-code even shorter, we introduce the commands Repeat and AddZOffset. For including the same GWL-file multiple times, the Include command can be combined with the Repeat command giving the number of repetitions. In our example, the include is listed only once but will be then repeated 10 times. This results in a total of 11 layers. Note that the included file must be adapted accordingly: the ZOffset must be incrementally increased with the command AddZOffset within the included script, instead of setting an absolute value in the main file. AddXOffset and AddYOffset work respectively for the x- and y-coordinates. We thus apply the following changes on *square1.gwl* and save it as *square2.gwl*:

```
0       0       0
10      0       0
10      10      0
0       10      0
0       0       0
write
AddZOffset 0.3
```

Listing 7.4: Square 2

We now change our *include.gwl* script accordingly, as in figure 7.3.

We are now able to program any 3D structure with a few basic GWL-commands. Read on to the next section 7.1.4 for how to repeat a structure several times on the same substrate with changing parameters.

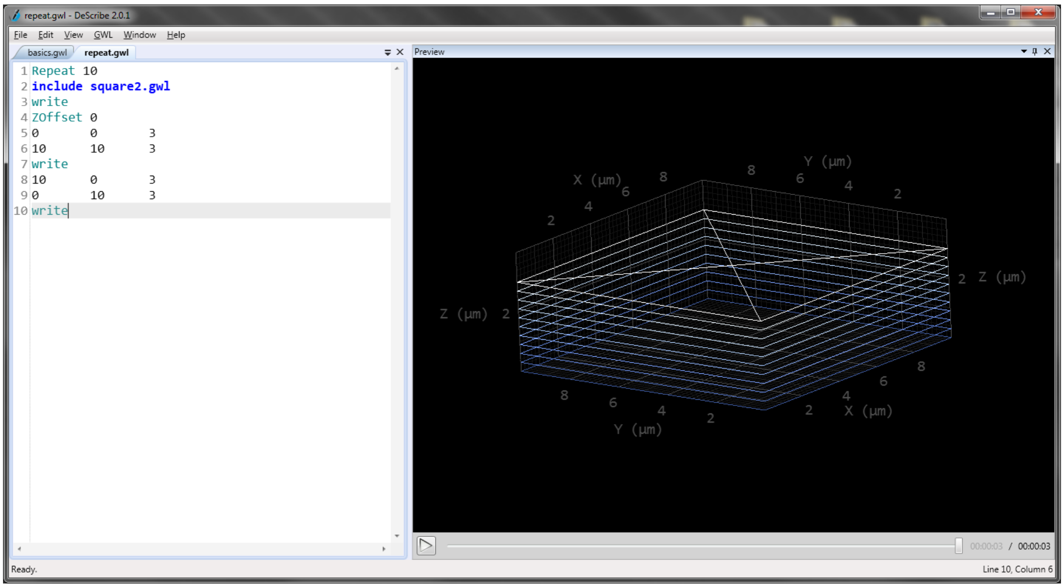


Figure 7.3.: DeScribe screenshot of the improved 3D example using `Include` combined with `Repeat`.

7.1.4. Repeating structures with changing parameters: 'Dose test'

In general when programming a new structure we do not know which parameters will yield the best results. In this section we learn how to define a dose test for finding the optimal laser power for the given experimental conditions. For more details about intensity and dose, please refer to section 7.4.

Since the dose influences important parameters of the writing process, like degree of polymerization and size and aspect ratio of the voxel, we need to make sure that we find the right dose for your structure. We recommend to start by defining the `ScanSpeed` you will use for your structure. Here you will typically face a trade-off between structure quality and writing time. For the Piezo, typical speeds are 25-300 µm/s. Considering the Galvo with the 63x NA1.4 objective recommended speeds are between 1,000 and 20,000 µm/s.

For a proper dose test we repeat the structure or a part of the structure several times at different locations on the substrate with varying laser powers. If you see bubbles in the live view, your dose is too high. Depending on the lateral structure dimensions the displacement between the structures can be performed either by the Piezo or by the stage.

7.1.5. Offsets

For the structure *include.gwl* a displacement using the Piezo is sufficient, since we're easily staying within its addressable volume. We are thus using the `XOffset` and the `YOffset` commands in the `PiezoScanMode` to write the structure in a 2×2 array with changing `LaserPower`. Note that a dose test can also be performed by changing the `PowerScaling` value. For a shorter code we replace the diagonal lines in the programming code with the comment *%add diagonal lines here%*. You can see the preview of a dose test in figure 7.4.

We now have a 2×2 array of the structure *include.gwl* written with increasing `LaserPower`. The dose test may now be evaluated using an optical microscope or a SEM and the best structure is picked or a finer dose test is performed in a second approach. Such a test must be understood as a very important step in Laser Lithography, since it is al-

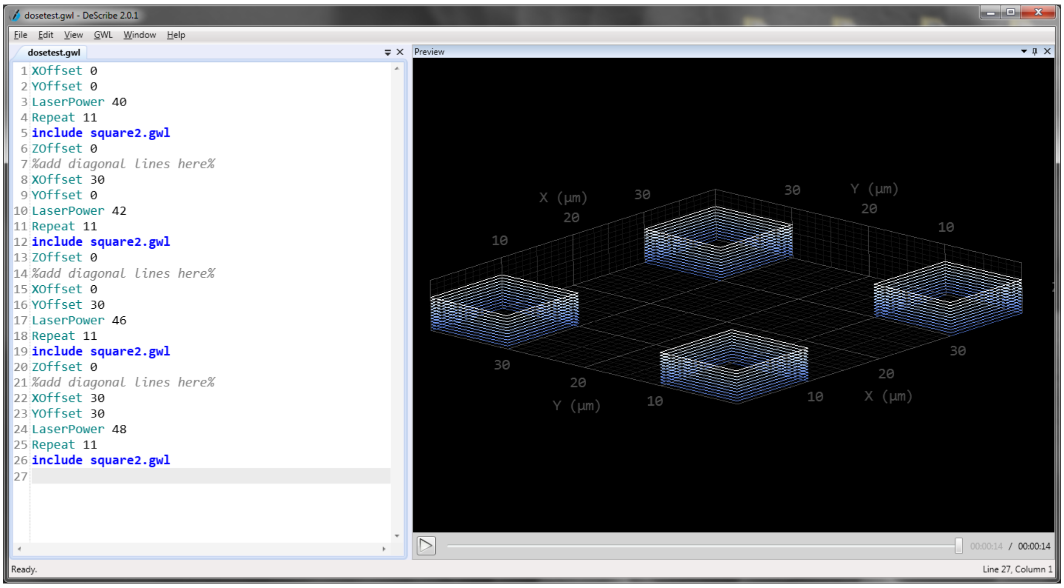


Figure 7.4.: DeScribe screenshot of a dose test performed for different laserpowers. The laserpower can be graphically represented using the **beam** button in the preview window.

ways difficult to know a priori the optimal `Laserpower` for a given structure in a given environment!

7.1.6. MoveStage

For larger structures a similar array can be fabricated using the `MoveStage` commands. This is also the preferred way of displacement for structures written with the `GalvoScanMode`. For simplicity we stick to include.gwl as unit cell and increase the lateral displacement so that the overall array exceeds the piezo range. Using the `MoveStage` command the lateral movement is performed by the stage and not by the piezo. Note that the displacement accuracy decreases accordingly. You can observe the preview of a dose test in figure 7.5.

Note that the Piezo coordinate system is relative to the current stage position. Therefore we can `Include` the same, identical GWL-file several times with constant x- and y-offsets but at different stage positions.

7.1.7. Include & file path

In general it is useful to separate point lists from settings and move commands. For this purpose include has already been discussed earlier in this chapter. The command is simply replaced by all the content of the file it points to. To give the location of the file there are three possibilities. Either you give an absolute, or a relative path or no path at all, in which case NanoWrite assumes the files in the same directory as the file that issues the include. All three variations are given in the following code example, where you will also see that a comment is indicated with a % sign.

```
%included file in current path
include pointlist1.gwl
%include file in subpath
include gwl-collection\pointlist2.gwl
%include anywhere on computer
```

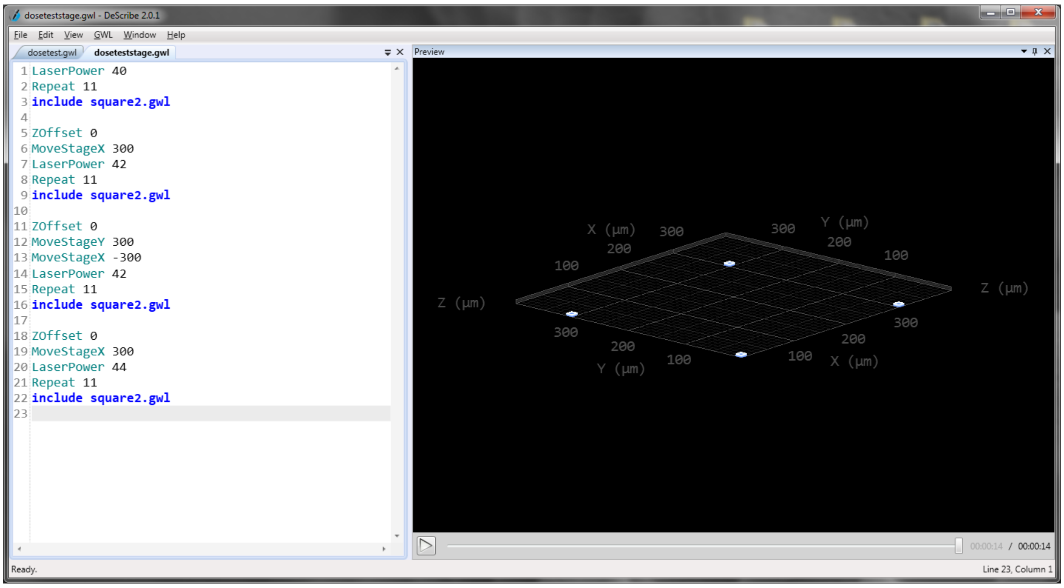


Figure 7.5.: DeScribe screenshot of a dose test performed for different LaserPower. On DeScribe, the laser power is not graphically represented. The stage was used for larger movements.

```
include d:\data\gwl\pointlist3.gwl
```

Listing 7.5: Different ways to include a file

7.1.8. Protocol & logging

For writing comments in the script that should be saved in a log file, the command MessageOut followed by the text sequence to save is available. For this purpose SaveMessages will save all previously appeared messages at the time the command is invoked. The default saving location will be the folder where the loaded GWL-file resides. The file name will be formatted like: *yyyymmdd-hhmmss_MessageLog.txt*. However you can also specify your own file name according to the same syntax as with Include, but note that the folder has to exist beforehand.

It is often useful to have the messages appearing on the right hand side of NanoWrite as a text-file on the computer. For this, the DebugModeOn must be activated. Using the message log it makes sense to also use TimeStampOn, which adds the current time according to the windows regional settings as a prefix to every message. TimeStampOn and TimeStampOff can be used anywhere in the code.

Furthermore it is possible to save the current live view image as a *tif* file with CapturePhoto. This command has to be followed by a file name with an optional path. The syntax is the same as with Include. The camera view within NanoWrite in the tab Camera has to be activated for this purpose. Pre-existing pictures with the same file name will be overwritten automatically. We recommend to set the camera to auto contrast and auto exposure for this feature.

7.1.9. Writing text

It is possible to write descriptive text into the resist using the WriteText command. The syntax requires quotation marks and line feeds can be invoked with \n. The classic example written in two lines would therefore be:

```
WriteText "Hello\nWorld"
```

With every \n the text left from the \n is shifted inverse to the distance defined by LineSpacingX, LineSpacingY and LineSpacingZ in µm. The effect is that the last line of a description, no matter how many lines it contains, stays always on the same position. If you use LineSpacingY>0 it is therefore important to add a sufficient offset with TextPositionY>0 to make sure we're still within the piezo range.

There are different options that can be set when using this command. With TextPositionX it is possible to set the x-coordinate of the description. TextPositionY and TextPositionZ function accordingly. The positioning commands work similar to the offset commands XOffSet, YOffSet and ZOffSet but are independent from them. The writing speed and exposure of the description can be set via TextScanSpeed and TextLaserPower, note that the normal LaserPower value has no influence on text writing. Finally with TextPointDistance the distance of interpolated points can be given in analogy to the normal PointDistance setting.

LaserPower & PowerScaling with WriteText
Please note that with WriteText the value of PowerScaling is always set to 1.0 and cannot be changed. This has no influence on normal structures.

7.2. Movements, offsets & stitching

This section covers all movement related commands. First of all, movements without exposure will be described. Then, we will have a closer look on how 3D structures are actually defined by moving the sample and exposing the resist. At the end we will talk about stitching and offsets.

7.2.1. Movements without exposure

In addition to the `MoveStage` command already introduced in chapter 7.1, some additional ways to move the stage and the piezo are available. They can be classified as absolute and relative movements.

Absolute movements

It is always possible to position the laser focus to a certain point within the writing volume without any exposure by giving its coordinates and setting the fourth value (=LaserPower) to 0, e.g.,

```
%Movements via coordinates and GotoX/Y
%are dependent on the ScanMode!
15    20    7     0
write
%go back with
0     0     0     0
write

%The same point can be addressed with
GotoX 15
GotoY 20
GotoZ 7
```

Listing 7.6: Absolute movements in general

However this way the current ScanMode defines how this point will be reached! If you need to specify directly whether the stage or the piezo will be addressed you have to use the following commands for the previously mentioned coordinates:

```
%We take the glas-resist interface as a reference
FindInterfaceAt 0
%and use the center of the sample as origin
CenterStage

%Piezo movements (ScanMode doesn't matter)
PiezoGotoX 15
PiezoGotoY 20
PiezoGotoZ 7
%reset piezo to origin
PiezoScanMode
0     0     0     0
write

%Stage movements (ScanMode doesn't matter)
StageGotoX 15
StageGotoY 20
%The stage has no z-axis -> the z-drive has to be used
```

```
%Note that the z-drive is always relative!
AddZDrivePosition 7
```

Listing 7.7: Absolute movements depending on ScanMode

If you would like to move on to the next sample position you can do this in an automated manner by entering `SamplePosition n` with n being one of the available positions according to your current sample holder, e.g., for the 10x30 holder that would be 1 to 10. Keep in mind that there has to be a sample that is appropriately prepared for the current objective and that this command triggers a subsequent approach.

7.2.2. Offsets

Considering the re-usability of already available point-lists, it is most useful to get familiar with the offset commands. Without the need to change the actual code they facilitate parameter searches and overall changes to the coordinates. Let's consider a real world example. Say you want to find out at which height a structure loses adhesion to the surface. You might know the approximate voxel height and the rough position of the interface, however there is still enough variation that needs an experimental approach. If we assume that a 1 µm tall voxel is roughly centered on the surface using the `FindInterfaceAt` command, i.e. half of its height is visible, then we need to write it at least half the voxel size higher for it to lose the connection to the surface. A possible GWL-file needed to investigate this could have following form:

```
%Header
PiezoScanMode
PulsedMode
ConnectPointsOn
ExposureTime 50
PowerScaling 1.0

FindInterfaceAt 0
XOffSet 0
YOffSet 0

%Main-Code
ZOffSet 0
XOffSet 0
Repeat 5
include point-list.gwl
%end of main gwl-file

%point-list.gwl
%here is your structure that starts at z-coordinate 0!
10    12    0    100
write
AddZOffSet 0.2
AddXOffSet 3
```

Listing 7.8: Finding the correct Z-Offset

The output of this code example is sketched with a single voxel in figure 7.6 in the upper image.

Note that negative z-offsets are only useful, if the coordinates have higher z-values. Refer to section 7.2.3 for an example of negative z-offsets.

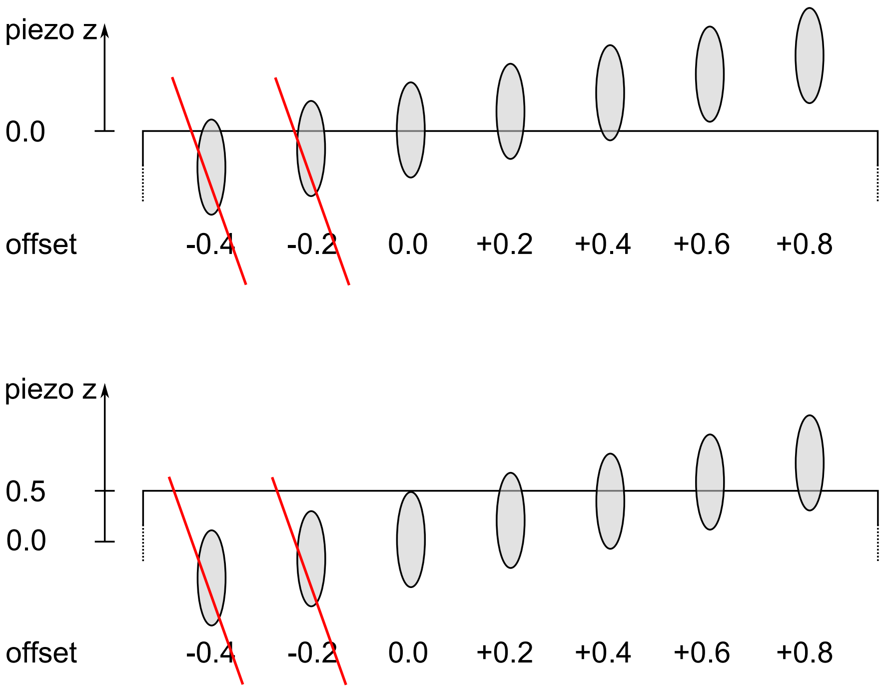


Figure 7.6.: This figure denotes the usage of offsets with coordinates that have $z = 0$. The upper drawing is valid for a `FindInterfaceAt 0`. For the lower sketch `FindInterfaceAt 0.5` has been used. For both a voxel height of 1 µm has been assumed. Note that the piezo does not accept negative coordinates, therefore the first two voxels are not valid!

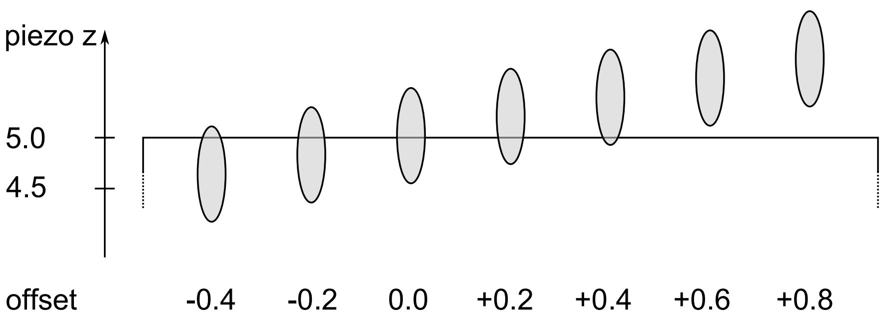


Figure 7.7.: This figure denotes the usage of offsets with coordinates that have $z = 5$. With `FindInterfaceAt 5` negative z-offsets have become useful. Example with second voxel from left: If the interface is at 5, the coordinate (12 11 **5**) and the `ZOffSet -0.2` then center of the voxel will be about 0.2 µm below the interface.

7.2.3. Basic Stitching

Stitching in x-y-plane

If you'd like to write a 500 × 300 µm^2 rectangle with the piezo, that is with PiezoScanMode, you would need to separate this task into three processes.

1. Write as much as possible within one piezo range
2. Move the stage such that the next piezo range is alongside the former one
3. Continue your structure with piezo movements as in the first step

Say you want to write it like this:

```
%StageScanMode is needed because the coordinates
%exceed the piezo range
StageScanMode
0       0       0
0       300     0
500     300     0
500     0       0
0       0       0
write
```

Listing 7.9: StageScanMode no stitching needed

Since 500-0 µm is bigger than the piezo range of 300 µm this code would only work with StageScanMode. However, if you change your code into this

```
%use PiezoScanMode for stitching
PiezoScanMode
300     0       0
0       0       0
0       300     0
300     300     0
write

MoveStageX 300
0       300     0
200     300     0
200     0       0
0       0       0
write
```

Listing 7.10: Stitching in PiezoScanMode

Of course it is also possible to stitch in y-direction and in both directions within the same job. The same basic principle applies.

Stitching along z-axis

The same principle is valid for vertical stitching. However, if you write higher structures the coarse movement will be executed by the z-drive of the objective instead of the piezo stage. Note that you need to implement DefocusFactor manually if you're not using the DiLL objective.

Let's assume you converted a 800 µm high structure from STL to GWL. You will need a header file with your preferred settings as usual. Then the DeScribe output file has to be adapted similar to the example in the last section, but this time with the command AddZDrivePosition instead of MoveStageX/Y.

```
%Start of structure
%as given by, e.g., DeScribe
10    10    0
10    20    0
...
10    10    300     0
10    20    300     0
write

%inserted manually
AddZDrivePosition 300
AddZOffset -300

%code continues
10    10    300     1
10    20    300     1
...
10    10    600     0
10    20    600     0
write

%again inserted manually
AddZDrivePosition 300
AddZOffset -300

%rest of code follows
...
```

Listing 7.11: Stitching along z-Axis

Note!
Vertical stitching without the DiLL-objective requires manual accounting for the DefocusFactor when using the AddZDrivePosition command.

7.3. Writing Modes

In this section we will present the different writing modes of the Photonic Professional (GT). A writing mode is the combination of a scanning mode, defining which positioning system is used, and an operation mode, defining how the laser beam is operated.

The Photonic Professional (GT) features three x-y-scan modes, corresponding to the three positioning systems:

1. `PiezoScanMode` for high resolution writing with a positioning accuracy in the range of 10 nm within a writing area of 300x300 μm^2. This represents the standard scanning mode for systems without the GT option (section 7.3.1).

2. `GalvoScanMode` for high speed writing. Positioning accuracy and writing area depending on the objective in use. This represents the Photonic Professional (GT) standard scanning mode (section 7.3.2).

3. `StageScanMode` for coarse writing with a positioning repeatability in the range of 1.5 µm on a large area defined by the maximum writing area on the current substrate (section 7.3.3). It is used for stitching large objects which exceeds the piezo or galvo scanning range.

The z-axis movements can be performed either with the piezo z-axis or with the z-drive of the microscope. The combination of galvo x-y scanning and microscope z-drive allows for continuous 3D writing. The combination of galvo x-y scanning and piezo z-axis movement is not synchronized, hence, e.g. a diagonal line in the x-z-plane needs to be written as a stair with alternating writing and z-moving commands.

The aim of this chapter is to present the strengths and weaknesses of each of these scan modes, allowing you to choose the best suited for your application. In addition to that, Photonic Professional (GT) features two operation modes, corresponding basically to the type of exposure on the programmed coordinate array:

1. `PulsedMode` for exposing individual points.

2. `ContinuousMode` for exposing continuous lines along trajectories defined by the programmed coordinates.

7.3.1. Piezo Writing

This scan mode is applicable for systems with and without the GT-option. It is based on a FBMS approach, meaning the complete sample holder is moved over the fixed laser beam in order to define structures. Since the center of the objective lens is used exclusively and the Piezo stage has a high linearity throughout its range this mode yields constantly accurate results within the full writing volume of 300x300x300 μm^3.

Both `ContinuousMode` as well as `PulsedMode` can be used. Especially for the latter it is important to understand that the Piezo goes through a transient state that takes between 200 and 400 ms. Thus a `SettlingTime` of 300 ms is recommended for most cases in order for the Piezo to reach steady-state.

In contrast to GalvoScanMode, tilt correction can be used with the Piezo, see 7.5.4.

Please refer to section 6.4.3 for details about using the `PiezoScanMode` with Piezo PerfectShape and 7.8 for writing without it, i.e. in constant speed mode.

7.3.2. GalvoScanMode

`GalvoScanMode` laterally uses the fast beam scanning option via a galvo scanner and vertically the z-axis of the piezo or the z-drive. This mode is ideally suited for layer by layer fabrication of large surface and large area structures.

Scan Field

In `GalvoScanMode` the diameter of the scan field and thus the (circular) fabrication area depends on the specific objective used. For lower magnification objectives the fabrication area is larger than for higher magnification objectives. The lateral writing range is listed in table 7.1 and compared in figure 7.8 with the piezo writing field.

Table 7.1.: Scan field as function of the used objective

Objective designation	Scan diameter	Squared scan field
63xNA1.4	$d = 200\,\mu m$	$a = 141\,\mu m$
25xNA0.8	$d = 400\,\mu m$	$a = 283\,\mu m$
20xNA0.5	$d = 600\,\mu m$	$a = 424\,\mu m$

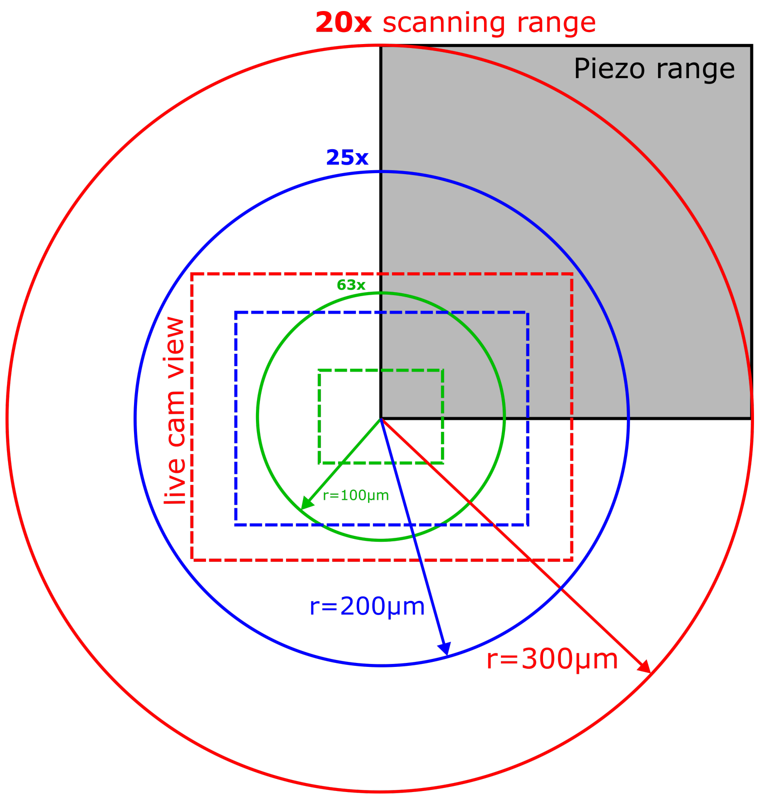


Figure 7.8.: Schematic to compare the writing fields and field of views (camera) for the 63x, 25x and 20x objective in the galvo mode with the writing field of the piezo mode.

In contrast to the `PiezoScanMode`, the origin of the coordinate system is located at the center of the scan field. As a result, your coordinates should not exceed $\pm d/2$ on axis or for arbitrary positions $\pm a/2$ respectively.

```
PiezoScanMode
0       0       0
300     0       0
300     300     0
0       300     0
0       0       0
write

GalvoScanMode
-70     -70     0
 70     -70     0
 70      70     0
-70      70     0
-70     -70     0
write
```

Listing 7.12: Mapping the scan field in PiezoScanMode and GalvoScanMode with the 63x immersion objective

The same list of coordinates can be written either in PiezoScanMode or in GalvoScanMode by using alternatively some offsets as long as the coordinates remain within the respective writing fields/volumes.

```
GalvoScanMode
XOffset  -70
YOffset  -70
0       0       0
140     0       0
140     140     0
0       140     0
0       0       0
write
```

Listing 7.13: Mapping the Galvo scan field through offsets

Step Size

Along with the fabrication area also the minimum step size (r) rises with lower magnifications. The beam scanner has a minimal angular step limiting the projected focus displacement at the objective focal plane. Using higher magnification objective therefore increases the quality of the path discretization. However the actual scan speed is respectively smaller. The scan speed limit is set by the angular velocity of the beam scanner.

ScanSpeed

Independently of the objective choice, the scan speed is simply defined through the command ScanSpeed. The value is given in µm/s as for in PiezoScanMode, however the typical range of the GalvoScanMode amounts to mm/s constituting a tremendous advantage over piezo writing.

PiezoScanMode versus GalvoScanMode
The most significant difference between these two scanning modes is the scan speed. Galvo writing allows for a scan speed of approximately two orders of magnitude higher than with the piezo. Additionally the required `SettlingTime` is essentially 0 ms[1] compared to 200-400 ms for the Piezo. In `GalvoScanMode` with `ContinuousMode` the movement is not interrupted and therefore defining a `SettlingTime` is not necessary. In terms of GWL programming it is also much simpler. As no velocity adaptation algorithm is necessary, `PerfectShape` has no need for configuration, and `ConnectPointsOn` is mainly ignored.[2]

Operation mode and writing strategy

Galvo writing needs a slightly different approach towards 3D Laser writing than piezo writing. Two elements account for this: the z-direction is not synchronized with the planar directions (X and Y) and the scan speed within a layer is significantly increased. In this section, we will analyze the implications on the choice of operation mode, the average scan speed of a whole layer and the control of the z-movements.

The two operation modes are compatible with `GalvoScanMode`: `ContinuousMode` and `PulsedMode`. The first is a horizontal vector writing mode and is recommended for most layer-by-layer structures. The latter is used when writing structures with very short line segments, or even points, in each layer. Trajectories with z-component can be sliced and fabricated using `PulsedMode` in `GalvoScanMode`.

Galvo PerfectShape

In order to reduce the effective writing time in `GalvoScanMode`, a smart algorithm has been integrated into NanoWrite 1.8. In `ContinuousMode` of `GalvoScanMode`, two consecutive and in-plane polylines are connected by a continuous and smooth trajectory increasing overall speed and trajectory accuracy. This new algorithm is called 'Galvo PerfectShape'. There are no user-defined settings and it doesn't need to be turned on or off. The combination of `GalvoScanMode` and `ContinuousMode` is always switched on.

PerfectShape
The green PerfectShape indicator in the NanoWrite user interface is on for both the Galvo and Piezo PerfectShape algorithm. Please note, that these two algorithms are completely different in the way they interact with the hardware. Merely their goal, higher speeds and higher accuracy, are the same.

Displacement in z-direction

For the displacement along the z-axis the Photonic Professional (GT) offers two possibilities: Piezo or microscope z-drive movements. The z-coordinate is fixed at the first

[1] values between 0 and 5 ms are recommended for `PulsedMode`.

[2] In `PulsedMode` the commands `ConnectPointsOn` and `ConnectPointsOff` are used for placing intermediary points between coded points with a defined `PointDistance`. However, this does not influence the scan speed.

z-coordinate of each trajectory. Note that the piezo z-axis and the microscope z-drive are much slower than the Galvo x-y-axes. This is why there is no vertical vector writing mode in `GalvoScanMode`.

- Piezo-displacement: The z-coordinates are addressed via the piezo z-axis. Before starting a layer the system waits for the defined `PiezoSettlingTime`. This allows the piezo to reach its steady state. The maximum scan range of the piezo in z is 300 µm. For higher structures either use vertical stitching with the microscope z-drive, or use the latter for displacement.

- Microscope z-drive displacement: The z-coordinates are addressed via the microscopes z-drive. The precision of this method is reduced compared to the piezo precision. In most cases this can be neglected if the movement is strictly monotonic. The scan range with this method is not limited by the scan range of the drive, but by the distance between sample surface and upper motor stage **surface**. (depending on your holder, between 1 and 3 mm). Higher structures will be ripped off when pulling on the sample holder. This method does not suffer from any stitching errors, the reduced precision is basically spread over the complete scan range. In general this method is favorable for layer by layer fabrication of structures higher than 300 µm.

Furthermore, due to the lack of a fast z-axis TiltCorrection is not supported in `GalvoScanMode`. `TiltCorrectionOn` and `TiltCorrectionOff` are therefore ignored by the system.

Vignetting and Distortion

With increasing radial distance, two errors gain importance: Vignetting and distortion. Both are relevant in x- and y-direction only and are not compensated for by the software of the system.

- Vignetting: The magnitude of the vignetting effect depends on the objective in use. It basically lowers the effective dose as a function of the radial distance to the center of the writing area. The smaller the fabricated structures, the smaller the effect of vignetting.

- Distortion: The value of the distortion effect depends on the objective in use. The effective position of the laser beam deviates from the programmed position. The larger the radial distance to the center of the writing area, the larger the distortion. The distortion plays a role when stitching different galvo writing fields, since the stitching error not only depends on the precision of the lateral displacement of the stage, but also on the maximum distortion at the edges of the galvo writing field.

7.3.3. Stage Scan Mode

The motorized stage allows for coarse displacements over large distances (few centimeters) along the x- and y-axes. Each sample holder is registered in NanoWrite with a given number of sample positions as well as their geometry. Each sample position center defines the origin of the x-y Cartesian system of the stage on the respective sample position. In `StageScanMode` the programmed point lists are addressed in this coordinate system.

Movements in z-direction are performed by the z-axis of the piezo. The main purpose of the stage is stationary positioning. Due to hardware, speeds beyond 300 µm/s will usually lead to strong vibrations which make useful writing impossible. If you are looking for high speeds, consider using the `GalvoScanMode`.

To activate the stage scanning mode, use the command `StageScanMode`.

Please note that the accuracy and repeatability of the Piezo as well as the Galvo are much better than those of the stage and we therefore recommend to use `Galvo- or PiezoScanMode` whenever possible.

Operation modes in StageScanMode

There are two operation modes for the `StageScanMode`: `PulsedMode` and `ContinuousMode`, which work in the same manner as in `PiezoScanMode`. However, note that the stage behaves differently than the piezo in a way that allows for vector movements instead of sequential movements: speed is not affected by distance to go through, nor will any update rate be used by the controller. This simplifies programming since only one single command is necessary for controlling velocity. Therefore, specify the `ScanSpeed` followed by a value in $\mu m/s$. In other words `UpdateRate` is of no concern for the `StageScanMode`.

Stage movements do not require any settling time. However, the vertical movements are operated by the piezo and therefore are subject to transient state behavior. Thus the value only applies to beginnings of curves with a changed z-coordinate. The `SettlingTime` is given in milliseconds. A value of 300 ms is advised. `ConnectPointsOn` and `PointDistance` only matter with `StageScanMode` in combination with `PulsedMode`.

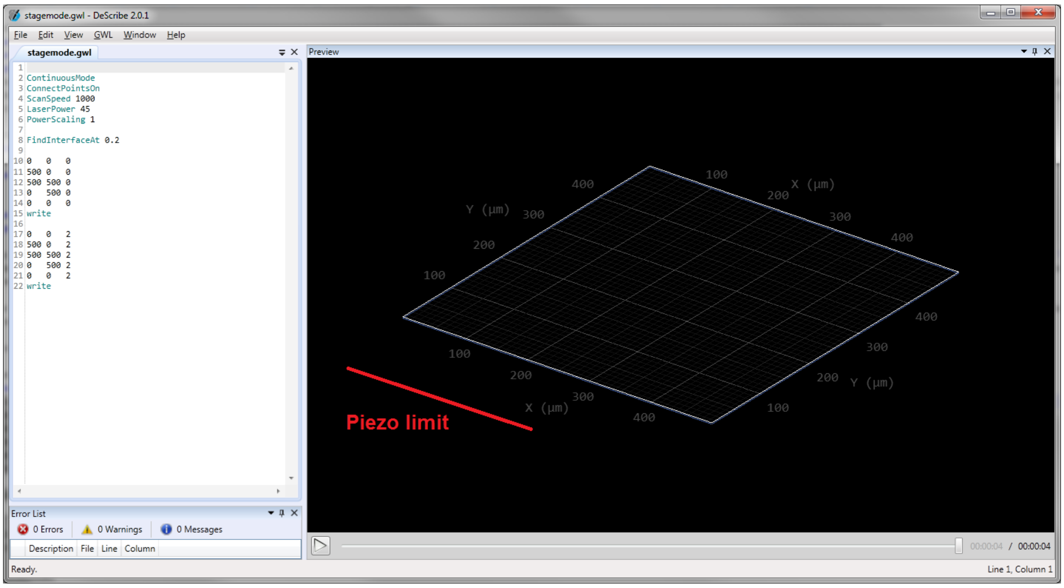


Figure 7.9.: DeScribe showing a 500 µm square printed in `StageScanMode`. As indicated by the red bar, the area of the piezo is limited. Therefore, this would not be possible to print in `PiezoScanMode`. Note that DeScribe does not produce any error message if the structure exceeds the piezo range in `PiezoScanMode`.

In the following an example GWL code is given for writing a structure with the stage movement which is bigger than 300 µm. The preview is visible in figure 7.9.

```
StageScanMode
ContinuousMode
ScanSpeed 200
LaserPower 45
PowerScaling 1.0

FindInterfaceAt 0.2

0      0      0
500    0      0
500    500    0
0      500    0
0      0      0
write

0      0      2
500    0      2
500    500    2
0      500    2
0      0      2
write
```

Listing 7.14: 500 µm square in StageScanMode

The StageScanMode is well adapted to write big structures with low required precision. Finer structures within coarse ones can be written in the PiezoScanMode. Substrate structuring can be a good application for stage writing.

7.4. Laser Intensity and Dose

This section deals with the necessary details of laser power, power scaling, laser intensity and dose within the resist.

Intensity

The laser device's output power is not changed during operation. For stability purposes it is kept at its maximum output power. The laser power that reaches the objective and results in a certain intensity distribution in the focal spot/volume is controlled via a voltage applied to an acousto-optic modulator (AOM) in the range of 0-5 V. Hence, 'laser power' typically refers to is the power that the AOM lets through reaching the objective.

This voltage is set by two commands and is defined as proportional to their product `LaserPower` ∗ `PowerScaling`. While `LaserPower` is limited to 0 ... 100 and given in %, `PowerScaling` is a factor from 0 to the maximum your system is able to deliver (see AOM calibration curve window). With `PowerScaling` on *default* = 1 and the maximum `LaserPower` value of 100, the system is calibrated to a certain mean power value entering through the aperture of the default objective. Refer to table 7.2 for the two possible variants. If you want to use higher power values than `LaserPower 100` you have to increase `PowerScaling`. In the end, it is up to the user which command is used for varying the intensity.

Table 7.2.: power calibration

System	Objective	Aperture	Average power
Photonic Professional	100x NA1.4	4.6 mm	20 mW
Photonic Professional (GT)	63x NA1.4	7.3 mm	50 mW

Dose

The laser power, i.e., the power that reaches the objective only determines the intensity within the focal spot. The dose, however, is the combination of time and laser intensity. Hence, how many photons are accumulated in the focal spot.

This dose can be increased by increasing the laser power or increasing the time the focal spot remains at the same position, using either scan speed or exposure time. Note that the dose is roughly proportional to $\frac{Laserpower^2}{scanspeed}$.

7.5. Finding interfaces

7.5.1. The Interface Finder

This section discuses the commands and the process of interface finding. Interface in this manual always refers to a sudden transition between two materials of different optical density, i.e. there is a abrupt change in refractive index along the optical path. Usually this will be glass-resist or the other way round. But it can also be e.g. resist-silicon or air-resist. The Photonic Professional (GT) is equipped with a sophisticated system, called *Definite Focus*, that is capable of detecting most interfaces automatically.

Please make sure a sufficient refractive index difference $\Delta n = n_{substrate} - n_{resist}$ at about 830 nm exists. Table 7.3 shows the required index differences. It is required to ensure a sufficient back reflection of the interface finder signal. By using the combinations recommended in this user manual, you should always be able to find the interface automatically. Of course non-standard combinations may be used as well. In such a case, one can use a

Table 7.3.: Required refractive index differences with respect to the objective lens in use.

$\Delta n > 0.03$	for 100x NA1.4 and 100x NA1.3
$\Delta n > 0.05$	for 63x NA1.4
$\Delta n > 0.10$	for 25x NA0.8

coating process to adjust the difference in refractive index with a very small layer of, e.g. ITO, Al_2O_3 or gold. With such a coating, you should be able to use 170 µm thick cover slips in combination with IP-Dip. Without the coating, the interface would surely not be found automatically.

On the other hand, note that a large refractive index difference between resist and substrate leads to unwanted back reflections during exposure. Near the surface of the reflective substrates standing waves may induce strongly fluctuating exposure doses and increase the risk of bubble creation or under-exposure. For example, this can be seen on silicon substrates where a standing wave pattern at the edge of the final structure is visible in vicinity of the substrate surface.

You can trigger an automated focus search by giving the command `FindInterfaceAt` z, with z being the height in your GWL-coordinate system at which you would like your interface to be positioned. The upper part of Figure 7.6 shows the achieved result with one voxel and different offsets for the standard value $z = 0$. In its lower part, you can see the same example with $z = 0.5$. As the Piezo does not accept negative coordinates you will have to use $z > 0$ if you want to expose your structure lower than programmed. This can be useful for different reasons, one being that an increased 2D resolution is needed. In this case, lowering the voxel into the substrate (note that this is only figuratively possible, and mainly valid for glass substrates) will cause most of the voxel to vanish leaving merely the upper part which has a smaller line width.

7.5.2. Multiple Interfaces

Sometimes your structure will have more than one interface. The automated focus would in this case try to find the first occurring interface from the objective upwards. If you need to find another interface you will need to use the commands `InterfaceMax/Min`.

Note!
`InterfaceMax` and `InterfaceMin` cannot be used at the same time. Each time one of the commands is issued, the older threshold will be deactivated.

To use these commands with the correct value, the approximate strength of such an interface needs to be known. This can be read from the output of the FindInterface procedure in the message pane. The first value gives the strength of the interface and the second value is inverse proportional to the exposure time of the Definite Focus sensor. This is how you choose which interface the automated focus will search for:

1. Prepare your sample as usual and follow the standard workflow until you have approached the sample.
2. click **Find Interface** and note the values in the message pane. E.g. 50@90
3. Dial the focus wheel (see Figure 3.2) manually until you have reached your desired interface.
4. click **Find Interface** again and note the values. E.g. 200@99
5. Choose a strength and exposure time close to your desired values and type those into the GWL-Mini-Script in the Advanced Tab. Please note that the inverted exposure time has to be given with three digits. E.g. InterfaceMin 170.096
6. Check your result: Touch LoadPosition, click **Approach Sample** and then **Find Interface**. The laser focus of the Photonic Professional (GT) should now be set on your desired interface.

If you are satisfied with the result you can use InterfaceMax/Min with the acquired values in your GWL-files.

Note!
Once you have finished your job, you should remember to reset this setting with `ResetInterface`. Otherwise the next user might be wondering why his interface will not be found. It is therefore best practice to use ResetInterface in your standard header file to make sure you are using standard values.

7.5.3. Autofocus with the 63x NA0.75 air objective

Note, this option is not suitable for galvo scanning, as the liquid resist is applied on top of the glass slide and scanning through the glass slide deflects the beam path.

With the adjustment ring on the 63x NA0.75 objective, the z-position of the focus can be adapted for writing through different substrate thicknesses.

Center Corrected: Here it is not possible to use the Autofocus, thus finding the interface has to be performed manually as follows:

- Find the air-glass interface. It should be visible as a very large interface-signal on the interface finder (Fig. 7.10a, 7.10c).

- Move the Objective lens approximately 105 µm upward with the microscope dials (Fig. 7.10b, 7.10d). Now you should be within the resist. Check this by writing some lines without using the `FindInterfaceAt` command.
- Slowly lower the objective lens while writing lines, until no polymerization occurs anymore. Then you should be at the interface between glass and resist.
- Once the Interface has been found with this method, you can automate this procedure by finding the air-glass interface and using the command `AddZDrivePosition 105` where 105 has to be replaced by the previously measured position of the interface.

Substrate Corrected: The autofocus can in principle be used. However, it is necessary to use `InterfaceMax` to prevent the autofocus from recognizing the air-glass interface instead of the glass-resist interfaces. For some resists the autofocus signal might be too weak and instabilities in the autofocus procedure occur, which results in interface failed messages. In these cases the threshold amplitude in the Interface Finder window may be lowered or the amplitude filter may even be switched off (Fig. 7.11a, 7.11b). The autofocus system is now more sensitive to fake signals, but the real interface should be detected in a more repetitive way (Fig. 7.11c).

7.5.4. Corrections Features

Defocusfactor

`DefocusFactor` is used to counteract the shift of the focal point position due to the mismatch of the refractive indices of immersion medium, substrate and photo resist. Its value will be multiplied to every subsequent z-coordinate in the point list (after adding the `ZOffSet` value). Theoretical values can be derived by Snell's Law.

If exact heights are of major concern we recommend to measure the `DefocusFactor` manually. This can be done by writing two lines which are suspended between two walls and separated in z-direction by a given distance. If the coded distance is 30 µm and the measured distance 20 µm, then `DefocusFactor 1.5` needs to be set.

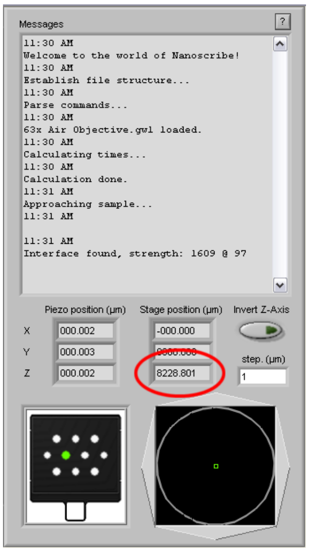
Please note that in dip-in configuration this factor is not necessary (i.e. DefocusFactor=1). This is due to the fact that there is no change of refractive index through which the laser transverses.

Tilt correction

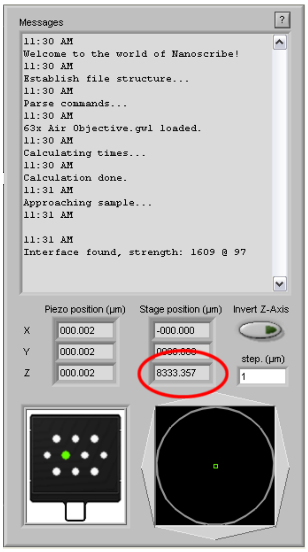
Defining structures along such an interface between a substrate and photo-resist usually suffers under a misalignment between the x-y-plane of the piezo and the interface plane defined by the substrate surface (see figure 7.12). This may lead to structures not having sufficient contact to the surface and thus being torn off during development. To compensate for a tilt of the substrate the coordinate system can be adapted by using the `TiltCorrectionOn` command. This command turns on the tilt correction feature with the last known angles in x- and y-direction. As a result a x-y-position dependent z-offset will be added to the user defined coordinates, i.e., the z-axis will not be rotated.

In order to measure the tilt on the current position you will have to issue `MeasureTilt n`, with $n > 2$ being the square root of the number of points where the z-position of the substrate will be measured. Recommended values are 4 to 6. Note that for `PiezoScanMode` these points will be evenly distributed throughout the $300 \times 300\,\mu m^2$ piezo range with one measurement at each corner. It is therefore highly recommended to reissue the `MeasureTilt` command whenever you move the stage in `PiezoScanMode`. For `StageScanMode` the measurements will take place over the complete addressable writing area on the current sample position (as defined in *Exchanger.ini*) and needs repeating only when changing the sample position. Based on the measured z-positions a linear fit will be calculated according to which the correction will be done. Note that it is possible to do a parabolic fit only in the user interface. Open tab **Advanced Settings** and click **Tilt correction**, where you can find the switch for the parabolic fit. In general after turning on tilt correction or executing a tilt measurement the `FindInterfaceAt` command should always be issued. Please note that with activated tilt correction the total addressable volume is reduced depending on the magnitude of tilt. Furthermore the coordinates displayed in the main window of NanoWrite reflect the corrected values, i.e., they will be different from the design coordinates.

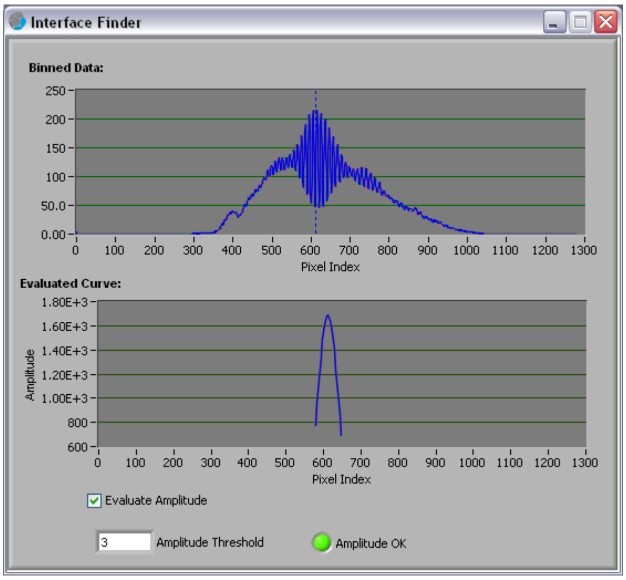
Note!
TiltCorrection does not work in GalvoScanMode due to hardware limitations.



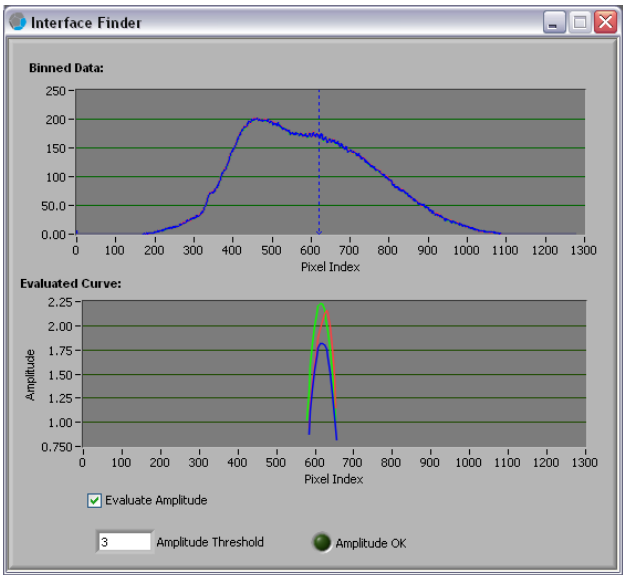
(a)



(b)

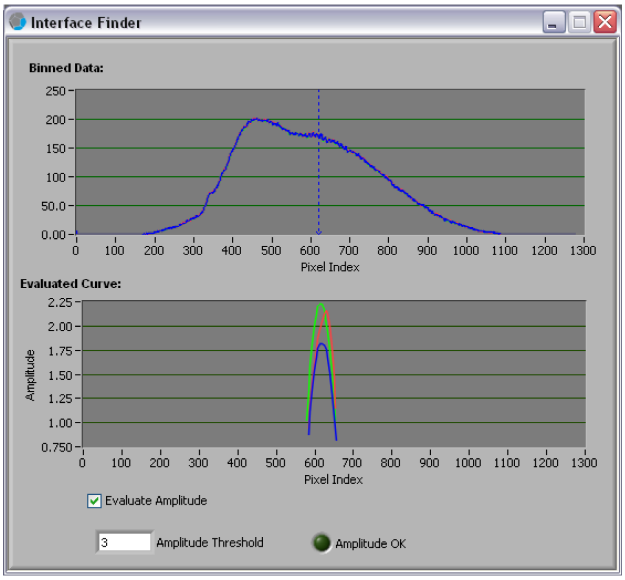


(c)

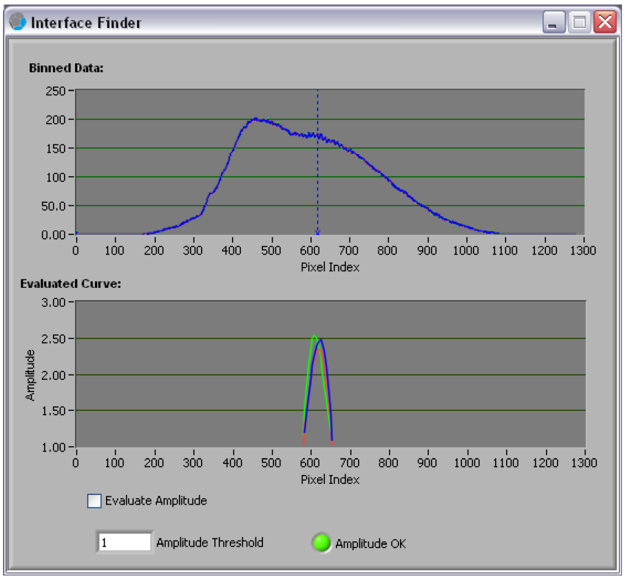


(d)

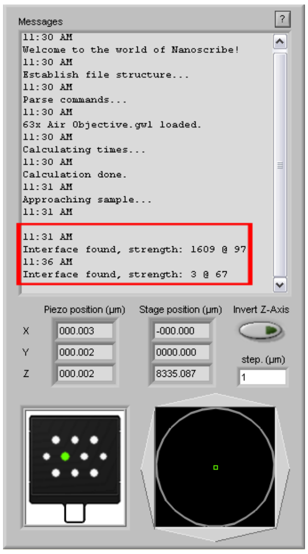
Figure 7.10.: NanoWrite main window at (a) air-glass interface and at (b) glass-resist interface. Interface finder window at (c) air-glass interface and at (d) glass-resist interface. Note that the glass-resist interface amplitude is too weak for being detected by the autofocus system with the current settings of amplitude threshold and evaluate amplitude.



(a)



(b)



(c)

Figure 7.11.: Increasing the sensitivity of the autofocus for detection of the glass-resist interface by lowering the amplitude threshold and disabling the evaluation of the amplitude. Autofocus finds the glass-resist interface with a lot lower strength as before (right picture)

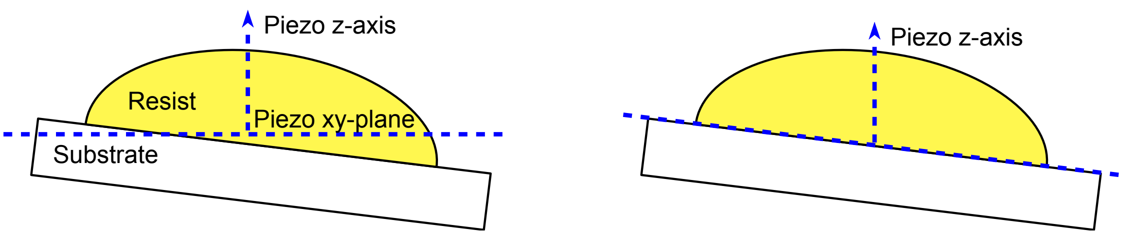


Figure 7.12.: This figure shows the principle of the tilt correction. To ease the understanding it has been reduced to a tilt in one direction only and the angle has been exaggerated. The left hand side shows the uncorrected situation, the right hand side demonstrates the tilt-corrected case.

7.6. Variables, Loops & Calculation

7.6.1. Variables

Before getting to loops, variables need a quick introduction. A variable is initialized with `var $var_name = value`. The variable name may contain capital letters, numbers and underscores. Its value must be a decimal number, but can be expressed as a mathematical term. Further, it can be referenced to it with `$var_name` and set to another value with `set $var_name = value2`.

Variables can also be defined locally, with the expression `local $var_name = value`. In this case, the scope of use is limited to the current script, and is not transferred through `include` command from a subscript.

If variables are global it doesn't matter if you define or change them in the top-level GWL file or in one of the included GWL files. The changes will be visible and identical at all levels. If you define a local variable in a file while this is also the name of a global variable defined in a parent file, the local variable has precedence over the global one. However, the latter remains unaffected in the remainder of the scripts and file hierarchy.

Note!
As a standard measure for debugging, you can use `ShowVar $variable` to print the value of the current variable to the **Message Log**. However, if you want more flexibility in displaying the variable value, string formatting options allow you to write it within a string sequence. Please refer to the following section for more details on string formatting.

7.6.2. String Formatting in NanoWrite

The string formatting gives access to the values of the variables, and use it automatically in combination with the commands `WriteText`, `MessageOut`, `LogModeFileName` and `CapturePhoto`. As a general feature, the current value of a given variable can be used as a string. Let's consider the following example:

```
Var $x=10
set $x=$x/3
WriteText "The value of x is %f" #($x)
```

Listing 7.15: String formatting

NanoWrite will automatically recognize the operator #, and write the following text: "The value of x is 3.333333".

- The % character initializes the format specifier which specifies the target, that is to say the position where the value is to be written in the text.

- The letter f sets the conversion code: f refers to float, d refers to signed decimal integer and u to unsigned decimal integer. The letters e and g can be used for scientific notations applied to floats. Note that for negative values, the conversion code u will result in 0.

- The operator # refers to the variable which value is to be taken. In our example the variable $x was modified, and therefore the last value was taken.

Table 7.4.: Optional syntax elements for string formating. In the example, we assume that the variable $x has the value 12.3456.

Element	Syntax	Example	Result
Signed integer	d	"x = %d" #($x) Var $z=-$x "z = %d" #($z)	"x = 12" "z = -12"
Unsigned integer	u	"x = %u" #($x) Var $z=-$x "z = %u" #($z)	"x = 12" "z = 0"
Float	f	"x = %f" #($x)	"x = 12.345600"
Float in scientific notation	e	"x = %e" #($x)	"x = 1.234560E+1"
Automatic float (Scientific notation only for exponents greater than 3)	g	"x = %g" #($x) Var $y=1000000*$x "y = %g" #($y)	"x = 12.3456" "y = 12.34560E+6"
Significant digits	[_n]	"x = %_3f" #($x)	"x = 12.3"
Precision	[.n]	"x = %.3f" #($x)	"x = 12.346"
Add a + sign	[+]	"x = %+f" #($x)	"x = +12.345600"
Minimal width > 0	[n]	"x = %12f" #($x)	"x = 12.345600"
Left justification	[-]	"x = %12d" #($x) "x = %-12f" #($x)	"x = 12.345600" "x = 12.345600 "
Fill with zeros up to minimal width	[0]	"x = %012f" #($x)	"x = 00012.345600"
Engineering notation (Exponent multiple of 3)	[^]	Set $x=1000000*$x "x = %^e" #($x) "x = %^.3e" #($x)	"x = 12.345600E+6" "x = 12.346E+6"

Some extra arguments can be given after the character % and before the conversion code in order to set syntax parameters. They are called modifiers and are optional. The different options for string formatting are presented in the Table 7.4. The optional modifiers are enclosed by brackets, although they must be written plain in a script.

Of course, string formatting is not restricted to the `WriteText` command: it can be used with `MessageOut` for writing the value of variables in the message log. Another application is in the `LogMode`, in which the command `LogModeFileName` is used for naming the log file, or with the `CapturePhoto` where a photo of the camera view is acquired, saved in the *.tiff format and named after a user defined text. This text can contain variables values.

Note!
If more than one variable is referred to within the text, the corresponding variables must be inserted after the # operator separated by comas. For example `#($x,$y)`.

7.6.3. Boolean expressions and loop

Boolean expressions

For enhancing the programming freedom within the GWL language, NanoWrite 1.8 introduces boolean expressions. The main application of booleans is their use in While loops and If-Elif-Else conditions. Boolean expressions can be defined using the standard relational operators “==”, “>”, “<”, “!=”, “=<”, “=>”, the unary operator `Not` and the binary operators `And`, `Or`. The relational operators have the highest precedence, followed by `Not`, `And`, `Or`. Complex boolean expressions are built using parenthesis. Boolean expressions cannot be used to assign values to a variable. However, they can be be built with existing variables. It is important though to remember that all variables are floats, and should therefore not be treated as integers. In this case, equality conditions might lead to unexpected results. For instance, the expression “1 == 1.00001” is indeed not true. This can be misleading if the second part of the expression is the actual numerical evaluation of a variable that was expected to be 1.

```

local $scanspeed = 10000
if $j > 15000
    WriteText " greater than 15000"
elif $j < 15000
    WriteText "Less than 15000"
end
```

Listing 7.16: If-Else

For and While Loops

In previous examples some loop commands have already been used to facilitate the understanding and shorten the code. Now we will introduce them decently. In GWL there are two different forms of loops. One, as you might be familiar with from other languages, is the for-loop. The other loop concept is much simpler, but also more restricted. The `Repeat n` command is used with a subsequent `Include` command and includes the given GWL-file $n + 1$ times. Note that it cannot be used for repeating any other command.

```
%Loops
%use standard header including
PowerScaling 1.0
XOffSet 0
local $i=0
local $start = 0
local $end = 3
local $increment = 1
for $i=$start to $end step $increment
   include square.gwl
   AddXOffset 100
   MultPowerScaling 1.1
end

%the same would be achieved with
Repeat 3
include square_withOffset&PowerScaling.gwl
```

Listing 7.17: For-loops

In this example both commands achieve the same result, however as soon as you try to implement something more advanced, e.g. where the loop variable will be used within the loop, the `Repeat` command will not suffice anymore. Also note that the second GWL-file has to include the offsets and power scaling commands. Thus it is not as easy to separate commands from point-lists when using `Repeat`. It should be noted that the prefix `Add` (`Mult`) is used to add the given value to the already existing setting.

Add & Mult
Note that the prefixes Add and Mult are viable for a number of commands. For details please refer to appendix B.

A local variable $k is used within a while loop. The boolean expression is tested at each iteration, until it returns False. In the loop, two conditional jumps are tested on the value of $k: if the first is true then the continue operates a jump to the next iteration, and if the second is true the break keyword exits the loop:

```
local $k = 0
while $k < 10
   set $k = $k + 1
   if $k == 2
      continue
   end
   if $k == 5
      break
   end
   TextPositionY 10*$k
   WriteText "New value %d" # $k
end
```

Listing 7.18: While-loops

7.6.4. Mathematical operations

As of NanoWrite version 1.7 some basic mathematical operations are supported. The syntax is straight forward. Trigonometrical functions assume radiants as argument. For a list of the implemented functions please refer to table 7.5. Since NanoWrite 1.8 the modulo function is also implemented. Note that it is based on floating point calculation. Take care not to use invalid input arguments, otherwise the output is not a number (NaN) and will result in wrong calculations! In the following example, a set of helices made with For loops is presented. You can see the preview in figure 7.13.

```
%This code creates a square array
% of helices using variables and for-loops.
var $twoPi = 3.1415926 * 2

%Helix parameters
var $r=1.9      % radius of helices
var $pitch=2    % pitch of helices
var $pitches=4     % number of pitches
var $a=5     % center distance of helices
var $footprint = 50      % footprint in um

%Loop variables: rotation angle and point coordinates
var $offx=0
var $offy=0
var $phi=0
var $x=0
var $y=0
var $z=0

%Position of first helix
```

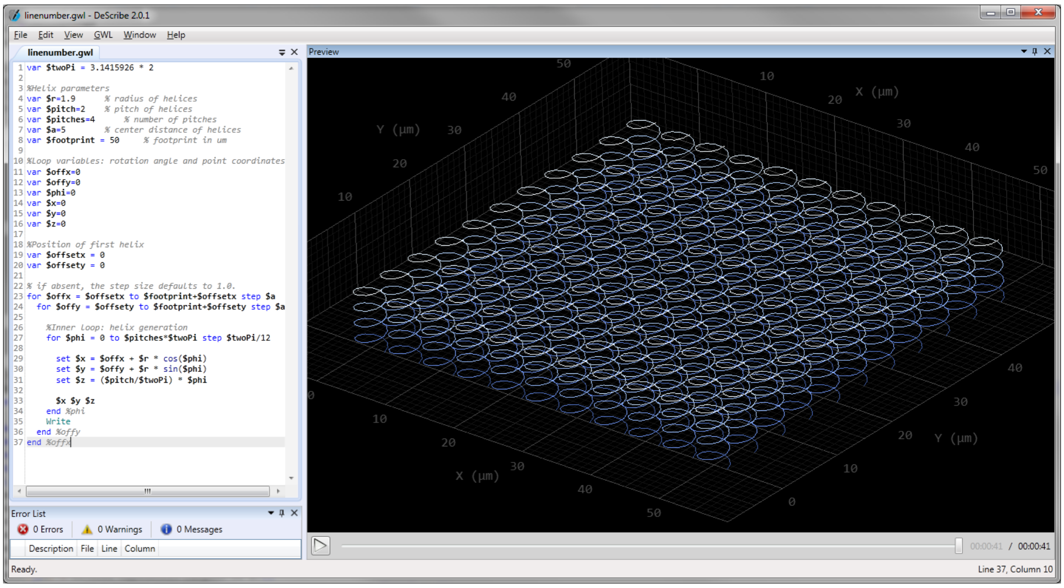


Figure 7.13.: View on DeScribe of a full 3D structure example. Helices are made out of their mathematical description. This is a complete 3D example since the three axis of the piezo stage move simultaneously.

```
var $offsetx = 0
var $offsety = 0

%Outer loops: helix placement via $offx and $offy
%Note: "step $variable" of for-loops is optional
% if absent, the step size defaults to 1.0.
for $offx = $offsetx to $footprint+$offsetx step $a
  for $offy = $offsety to $footprint+$offsety step $a

    %Inner loop: helix generation
    for $phi = 0 to $pitches*$twoPi step $twoPi/12

      set $x = $offx + $r * cos($phi)
      set $y = $offy + $r * sin($phi)
      set $z = ($pitch/$twoPi) * $phi

      %Emit computed coordinates
      %Note:
      % coordinates can be literal numbers or variable
      % identifiers but not mathematical expressions
      $x $y $z
    end %phi

    %Terminate polyline
    Write

  end %offy
end %offx
```

Listing 7.19: Square array example

Table 7.5.: This table lists the implemented mathematical functions in NanoWrite

Function	Description
abs($x)	absolute value of x.
acos($x)	inverse cosine of x in radians.
acosh($x)	inverse hyperbolic cosine of x.
asin($x)	inverse sine of x in radians.
asinh($x)	inverse hyperbolic sine of x.
atan($x)	inverse tangent of x in radians.
atanh($x)	inverse hyperbolic tangent of x.
ceil($x)	rounds x to the next higher integer (smallest integer ≥ x).
cos($x)	cosine of x, where x is in radians.
cosh($x)	hyperbolic cosine of x.
cot($x)	cotangent of x (1/tan(x)), where x is in radians.
csc($x)	cosecant of x (1/sin(x)), where x is in radians.
exp($x)	value of e raised to the x power.
expm1($x)	returns e^x-1.
floor($x)	truncates x to the next lower integer (largest integer ≤ x).
getexp($x)	returns the exponent of x in base 2.
getman($x)	returns the mantissa of x in base 2.
ln($x)	natural logarithm of x (to the base of e).
lnp1($x)	natural logarithm of (x + 1).
log($x)	logarithm of x (to the base of 10).
log2($x)	logarithm of x (to the base of 2).
sec($x)	secant of x, where x is in radians (1/cos(x)).
sign($x)	returns 1 if x > 0, returns 0 if x = 0, and returns –1 if x > 0.
sin($x)	sine of x, where x is in radians.
sinc($x)	sine of x divided by x (sin(x)/x), with x in radians.
sinh($x)	hyperbolic sine of x.
sqrt($x)	square root of x.
tan($x)	tangent of x, where x is in radians.
tanh($x)	hyperbolic tangent of x.
atan2($x,$y)	arctangent with two arguments
min($x,$y)	minimum of two values
max($x,$y)	maximum of two values
mod($x,$y)	remainder after division of $x by $y
max($x,$y)	maximum of two values
pow($x,$y)	calculates $x exponent $y
rand()	returns a random real number between 0 and 1

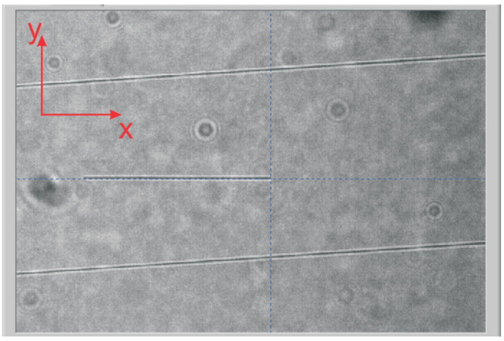


Figure 7.14.: Microfluidic channel and straight line along x in the default coordinate system.

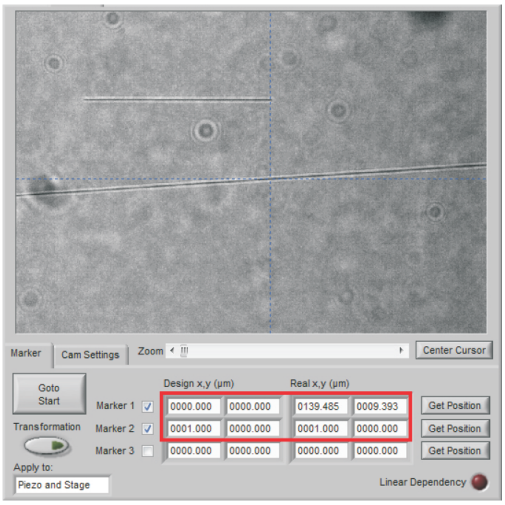


Figure 7.15.: Definition of marker 1. for marker writing

7.7. Marker Writing

This section explains how to align the designed structures with existing structures on a sample.

7.7.1. Alignment with two markers

To explain how the alignment with two markers works a microfluidic channel is considered which is installed inclined along the x-direction of the system. A line written along the x-axis of the piezo coordinate system will not be parallel to the microfluidic channel's side-walls as shown in Figure 7.14. The following explains step by step how to adjust the piezo coordinate system to the microfluidic channel by rotating the coordinate system so that the x-axis of the piezo is parallel to the sidewalls of the channel.

In order to rotate the piezo coordinate system in alignment to existing structures the direction of one arbitrary vector is redefined. This vector is given by two coordinates in the **Design x,y (µm)** control boxes (Figure 7.15). For simplicity, the x unit vector is used in this section. The new direction of the vector in the rotated coordinate system is then defined via addressing two arbitrary points, as **Real x,y (µm)**, in the writing area along the desired direction (Figure 7.15, 7.16). These points may either be addressed via a stage or a piezo movement. To rotate the structures knowing the desired angle of rotation, the rotation can be set directly within the camera tab of NanoWrite under *defined by*

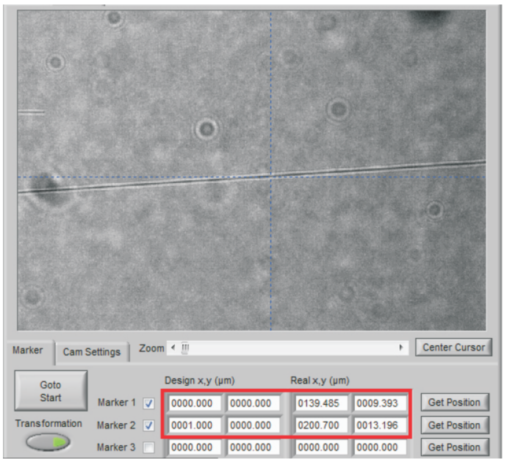


Figure 7.16.: Definition of marker 2. for marker writing

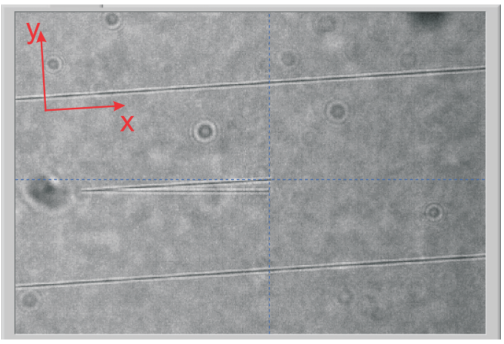


Figure 7.17.: Microfluidic channel and straight line along x and parallel to the channel in the aligned coordinate system.

First, the position of the laser beam has to be marked on the camera window. Therefore, the last written position is used as indicator of the laser position. This works of course only if the sample has not been moved in the meantime. This position is marked with the blue cross hair in the camera window by dragging the cross hair with the mouse (Figure 7.14).
Now, the new coordinate system can be defined. Move the structure via the stage or via the piezo so that the border of the channel is aligned with the cross hair as shown in Figure 7.15. Take this point as a first point of the new x-axis by clicking on click **Get Position** for marker one. The absolute coordinates on this sample for this point is then shown in the **Real x,y (µm)** boxes. Move the structure to an arbitrary second point on the edge of the channel and take this point as marker two by clicking on click **Get Position** for marker two (Figure 7.16). This defined the direction of the new x-axis. To activate the writing in the transformed coordinate system make sure the **Transformation** control is activated. The rotation angles of the x- and y-axis, as well as the deformation, are displayed in the message log. However there will be no deformation if the transformation is used with two markers only. No matter which distance the markers have, only the vector direction needs to be defined, not its length. Thus writing again the straight line along the x-axis will now result in a line parallel to the channel edges as shown in figure 7.17.

Note!
After transforming the coordinate system the starting point needs to be reset either to 0,0 or the first coordinate ("goto start") has to be set by moving the stage to the desired location.

7.7.2. Alignment with three markers

The alignment with three markers is based on the same principle as the marker alignment with two markers. Using three markers enables not only to rotate the coordinate system but also to stretch or shrink it or even to shear it. Choose three positions on the substrate with known coordinates, address them using the cross hair, enter the coordinates in the **Define x,y (µm)** column and get the absolute positions for each marker (by clicking on click **Get Position**). The rotation angles as well as the deformation factors of both axes will be visualized in the message log as soon as the **Transformation** is activated. Please remember that with 3 markers as well as with 2 markers, the starting point of the structures has to be reset after activating the transformation.

7.8. Piezo writing in constant speed mode

For users wishing to work with a constant speed in `ContinuousMode`, or to write in `PulsedMode`, the Piezo *PerfectShape* module is not suitable. *PerfectShape* can be deactivated with the `PerfectShapeOff` command. This does only apply for the Piezo *PerfectShape*. Galvo *PerfectShape* is always switched on when using `GalvoScanMode`.

7.8.1. ConnectPoints

ConnectPoints
When starting NanoWrite, PerfectShape and ConnectPoints are activated by default. Users wishing to work at constant speed must deactivate PerfectShape through `PerfectShapeOff`. As discussed in this section, ConnectPoints can also be deactivated by `ConnectPointsOff`. In this case, there is a risk of writing at dangerous frequencies for the piezo stage. As a result, it is advised to always insert the `ConnectPointsOn` in the script

To ensure a controlled speed and a controlled trajectory a line segment is automatically interpolated with intermediate points that are transferred one after the other at a specific `UpdateRate` *ur* to the piezo. This is performed through an automatic interpolation with which the distance *pd* of the interpolated points is set with the command `PointDistance`. Considering the following conditions:

```
PiezoScanMode
ContinuousMode
ConnectPointsOn
PerfectShapeOff
UpdateRate ur
PointDistance pd
```

Listing 7.20: `ConnectPointsOn` calculation example

The writing speed v will be constant: $v = ur \times pd$ in $\mu m/s$

In `ContinuousMode`, constant speed is particularly efficient when writing structures mainly composed of straight lines, such as gratings, woodpiles or photonic crystals.

In `PulsedMode` the automatic interpolation may be used to help programming periodic structures with just a few coordinate inputs. Note that the PerfectShape module does not apply to this operation mode.
In the following all four printing options are explained, two operation modes with and without interpolation:

ContinuousMode and ConnectPointsOn

`ContinuousMode` and `ConnectPointsOn` is the mode most often used. It is explained with the following GWL-code:

```
PiezoScanMode
ContinuousMode
ConnectPointsOn
PointDistance 100
UpdateRate 1000
```

```
0       0       0
1       0       0
write
```

Listing 7.21: `ConnectPointsOn` in `ContinousMode` example

Figure 7.18.: Illustration of the coded coordinates (black circles), the automatically interpolated coordinates (red circles) and the written result (blue line) in `ContinuousMode` and with `ConnectPointsOn`.

Figure 7.18 shows the coded coordinates (black circles), the additionally generated coordinates due to `ConnectPointsOn` (red circles) and the written result (blue line). In `ContinuousMode` the exposure is switched on at the first coordinate of the line segment and switched off at the last coordinate. The laser power is kept constant throughout the complete line segment apart from the line start and the line end where the laser power is adapted to the acceleration and deceleration behavior of the piezo. The points are addressed at an `UpdateRate` of 1000 Hz, the writing speed thus calculates as follows:
$v = ur \times pd = 1000\,Hz \times 100\,nm = 100\,\mu m/s$
The written result is thus a 1 µm long line written at the constant speed of 100 µm/s.

ContinuousMode and ConnectPointsOff

This is an advanced writing mode that should be used by very experienced users only who are familiar with the complete functionality of the system. The following GWL-code explains this option:

```
PiezoScanMode
ContinuousMode
ConnectPointsOff
PointDistance 100
UpdateRate 1000
0       0       0
1       0       0
write
```

Listing 7.22: `ConnectPointsOff` in `ContinousMode` example

Figure 7.19 shows the coded coordinates (black circles) and the the written result (blue line). In `ContinuousMode` the exposure is switched on at the first coordinate of the line segment and switched off at the last coordinate. The points are addressed at an `UpdateRate` of 1000 Hz, the writing speed thus calculates as follows: $v = ur \times pd = 1000\,Hz \times 1000\,nm = 1000 \mu m/s$

Figure 7.19.: Illustration of the coded coordinates (black circles) and the written result (blue line) in `ContinuousMode` and with `ConnectPointsOff`. Note that no coordinates are added when `ConnectPointsOff`.

The written result is thus a 1 µm long line written at the constant speed of 1000 µm/s. Note that since ConnectPointsOff was used the command PointDistance is neglected and the PointDistance *pd* for the calculation of the writing speed is the actual programmed distance (1000 nm in this code example). For such a short line this is still acceptable, but for longer lines the programmed speed increases significantly which may cause damage to the piezo. Furthermore, please note that the speed along a longer line segment with different point distances is no longer constant, but varies with the local point distance:

```
ContinuousMode
ConnectPointsOff
PointDistance 100
%This PointDistance command is ignored
UpdateRate 1000
0      0      0
1      0      0      % v=1000 um/s
1      10     0      % v=10000 um/s
100    10     0      % v=99000 um/s
100    10     0      % v=99000 um/s
0      10     0      % v=100000 um/s
write
```

Listing 7.23: ConnectPointsOff in ContinousMode (variable writing speed)

Hardware Danger!
Avoid driving the piezo at its resonant frequency (about 100 Hz) to avoid damages. Check the designs and make sure no periodically back and forth or circular movements close to 100 Hz or higher are programmed. Notice that the structure size and the writing speed as well as the actual structure design play a crucial role. We do recommend to stay below 10 Hz driving frequency.

PulsedMode and ConnectPointsOn

The following GWL-code explains this option:

```
PiezoScanMode
PulsedMode
ConnectPointsOn
ExposureTime 50
PointDistance 100
UpdateRate 1000
0      0      0
1      0      0
write
```

Listing 7.24: ConnectPointsOn in PulsedMode example

Figure 7.20 shows the coded coordinates (black circles), the additionally generated coordinates due to ConnectPointsOn (red circles) and the written result (blue dots). In PulsedMode the exposure is switched on and off separately at each point for the specified ExposureTime. The GWL-code thus generates 11 points at a distance of 100 nm.

Figure 7.20.: Illustration of the coded coordinates (black circles), the automatically added coordinates (red circles) and the written result (blue dots) in `PulsedMode` and with `ConnectPointsOn`.

PulsedMode and ConnectPointsOff

The following GWL-code explains this option:

```
PiezoScanMode
PulsedMode
ConnectPointsOff
ExposureTime 50
PointDistance 100 %ignored
UpdateRate 1000 %ignored
0     0     0
1     0     0
write
```

Listing 7.25: `ConnectPointsOff` in `PulsedMode` example

Figure 7.21 shows the coded coordinates (black circles) and the written result (blue dots). In `PulsedMode` the exposure is switched on and off separately at each point for the specified `ExposureTime`. The GWL-code thus generates 2 points at a distance of 1000 nm. The speed is not defined by the update rate but by the `ExposureTime` and the `SettlingTime`.

Figure 7.21.: Illustration of the coded coordinates (black circles) and the written result (blue dots) in `PulsedMode` and with `ConnectPointsOff`. Note that no coordinates are added when `ConnectPointsOff` is set.

7.9. Piezo PerfectShape Laser Power Adaptation

Due to inertia of the piezo the writing speed is not constant over a complete trajectory. The piezo needs some time to accelerate to the specified writing speed and some time to decelerate at the trajectory end. Thus the laserpower has to be adjusted at trajectory-starts and at trajectory-ends to avoid overexposure at the segments where the writing speed is slower than specified. This laser power adjustment is fully automated and integrated in NanoWrite. Since the exact parameter set for the laser power adjustment strongly depends on the properties of the piezo and of the photosensitive material used, the complete set of parameters used for the adjustment procedure is addressable via GWL-commands. In this section the complete parameter set is introduced. Further, it it explained how the adjustment takes place and how the parameters are optimized.

For a proper adjustment of the laser power at trajectory starts and ends, the exact times the piezo needs to accelerate and to decelerate has to be known. These times are defined via the GWL-commands `AccelerationTime` and `DecelerationTime` in seconds. Technically, to account for the `DecelerationTime`, the last point of a trajectory is addressed `UpdateRate`×`DecelerationTime` times. In general acceleration-times and deceleration-times are about 0.05 s. Furthermore the shape of the power adjustment curve has to be specified. The according GWL-commands are `Acceleration` and `Deceleration`. The parameter value is an exponent. The exact formula for the calculation is explained later in this section.

Finally, the `LaserPower` should not start and stop at 0 at the trajectory start and end respectively. The `LaserPower` should be at minimum of the exposure threshold of the material at very slow speeds. The command `Threshold` sets the exposure threshold in percent of the `LaserPower` at `PowerScaling 1`.

The `LaserPower` P at a trajectory starts as a function of the time t_{acc} elapsed since the start of the acceleration procedure calculates as follows:

$$P(t_{acc}) = \left(\frac{t_{acc}}{T_{acc}}\right)^{1/a} \times (P_s - P_t) + P_t \tag{7.1}$$

where T_{acc} is the `AccelerationTime`, a denotes the parameter `Acceleration`, P_t denotes the `Threshold` and P_s denotes the `LaserPower` set by the user.

The `LaserPower` P at a trajectory end as a function of the time t_{dec} elapsed since the start of the deceleration procedure calculates as follows:

$$P(t_{dec}) = P_s - \left(\frac{t_{dec}}{T_{dec}}\right)^{1/d} \times (P_s - P_t) \tag{7.2}$$

where T_{dec} is the `DecelerationTime`, d denotes the parameter `Deceleration`, P_t denotes the `Threshold` and P_s denotes the `LaserPower` set by the user. Figure 7.22 shows some examples for the power adjustment with different values for `Acceleration` and `Deceleration`. Following general parameters have been used:

```
LaserPower 30
Threshold 15
AccelerationTime 0.02
DecelerationTime 0.06
```

7.9.1. Adjusting the parameters

For a proper adjustment of the parameters use `LogMode` to log the effective positions of the piezo throughout one line. Following code may be used for this purpose:

```
OperationMode 2
PointDistance 200
UpdateRate 1000
```

The log file generated shows 8 columns that are indicating following values:

$x_{specified}$ $y_{specified}$ $z_{specified}$ *Laserpower* x_{real} y_{real} z_{real} *AOMvoltage*

Knowing the `UpdateRate` the piezo velocity can be calculated numerically from this log-file. Since the `UpdateRate` specifies at which rate the coordinates of one trajectory are transferred from the controller to the piezo, the velocity as a function of the displacement in x-direction $v(x)$ calculates as follows:

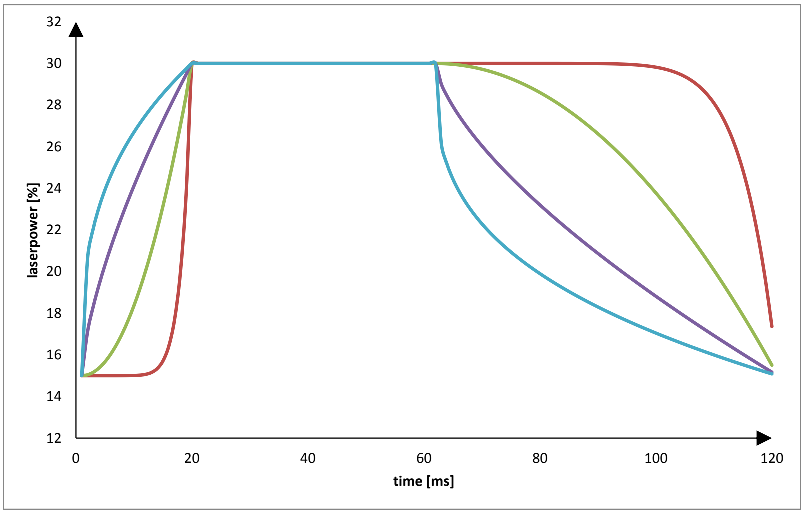


Figure 7.22.: Power adjustment for different settings of Acceleration and Deceleration (blue: Acceleration 3, Deceleration 3; purple: Acceleration 1.5, Deceleration 1.5; green: Acceleration 0.5, Deceleration 0.5; red: Acceleration 0.1, Deceleration 0.1).

$$v(x_n, x_{n-1}) = |(x_n - x_{n-1})| \times UpdateRate \tag{7.3}$$

PerfectShape Power Adaptation

This section is of interest for advanced users addressing two issues:

1. Polymerization is affected by the velocity variation, as the effective dose varies. A common problem is to observe lines with a non-constant width. At its extremities the piezo accelerates and decelerates, while in the center domain it reaches its maximum speed. The dose is then higher where the speed is lower, and therefore so is the line width. For compensating this effect a laser power adaptation algorithm is introduced for tuning the effective laser power to the current velocity.

2. When writing in the oil immersion configuration through a glass substrate, the optical index difference between the substrate and the resist induces de-focusing which is proportional to the distance of the focus to the substrate. From a practical point of view, a movement of the piezo in z direction is not equivalent to the corresponding movement of the focus in the resist, and the voxel is elongated as the focus is moved away from the substrate interface. The actual position of the focus is corrected by the `DefocusFactor` (Refer to 7.5.4 for detailed explanations) and the voxel elongation is balanced by laser power height adaptation.

The user has the possibility to choose and edit the laser power adaptation parameters depending on the experimental conditions. Its algorithm relies on the following equation, where the set laser power P_0 is multiplied by two functions:

$$P_{eff} = P_0 \times \eta(|v|) \times \mu(z) \tag{7.4}$$

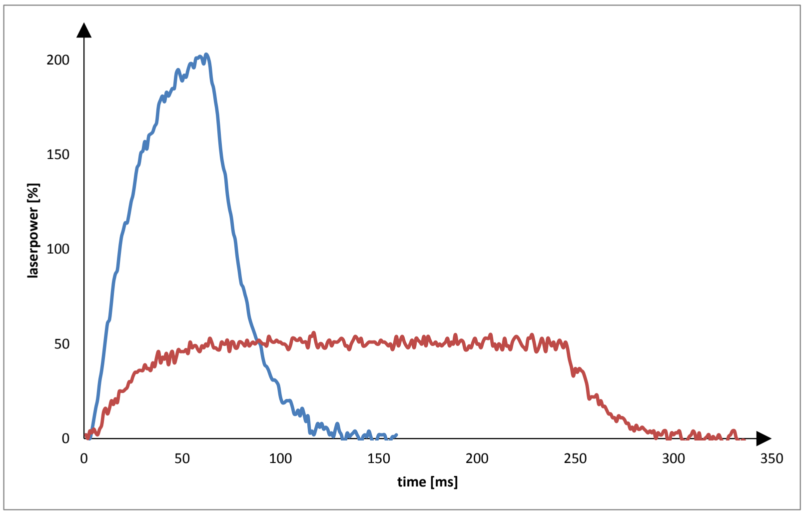


Figure 7.23.: Piezo velocity profile at 50 µm/s (red) and at 200 µm/s (blue).

The function $\eta(|v|)$ describes the laser power adaptation with respect to the scanning speed v. Numerical values are derived from an experimental (or user-defined) profile. Already present as a standard configuration, the "IP Resist" power profile is adapted to the IP-resists which can be procured through nanoscribe.

The function $\mu(z)$ implements a power adaptation over height. The corresponding command `psPowerSlope` sets the linear compensation parameter with respect to the z-position. In an oil immersion configuration, with the 100xNA1.4 oil immersion objective and IP-L780 photoresist, `psPowerSlope 0.025` is advised. Finding the right parameter for a given configuration (photoresist + objective) might require some additional testing from the user.

Fundamentally, power adaptation over height is required in presence of an optical interface glass-resist or air-resist. As a result, it needs to be deactivated with `psPowerSlope 0` when writing in the DiLL configuration, which is also the default setting.

Design personalized Power Adaptation profile

According to Figure 7.24, personal profiles can be created, from custom experimental data. This is relevant when working with other kind of resists than IP-resists. To create a custom profile, the voxel size variation must be measured as function of both the laser intensity and the piezo velocity. From this data, the velocity-power characteristics leading to a constant voxel size can be plotted.

The most direct way to operate is to write series of lines, while varying subsequently the `LaserPower` and the `ScanSpeed`. With a Scanning Electron Microscope the line width can be measured as function of speed and power. Make sure that the measurement are repeated to improve the statistics. Then the characteristics should be normalized with respect to the 100 μm scanning speed.

In NanoWrite, the Power Adaptation profile editor allows to create a new profile (Fig. 7.24). Two adaptation types are available: either a two-photon absorption model with threshold (valid for IP-Resists), or an interpolation mode. With the latter, it is required to

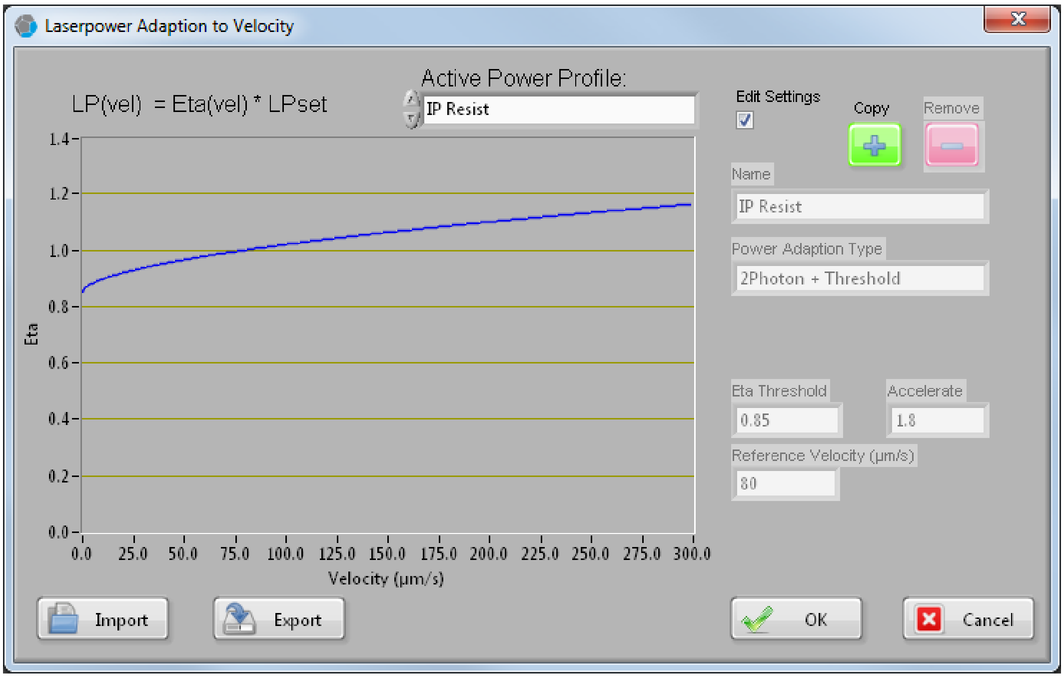


Figure 7.24.: NanoWrite menu for setting the active power profiles. The possibility to edit copies of these profiles is offered, as well as creating new ones. The graphics can be manually edited so that the curve fits to custom experimental results. To access this window, click **LaserPower Settings** in the window displayed in Figure 6.7.

enter a relevant number of points constituting the experimental characteristics. The $\eta(|v|)$ curve is interpolated, and is available in the drop down menu as a new profile.

The two-photon absorption model relies on an equation of the form:

$$\eta(|v|) = \eta_{th} + (1 - \eta_{th}) \times (\frac{v}{v_{ref}})^{1/acc} \tag{7.5}$$

Where η_{th} is the threshold for the laser power adaptation, v_{ref} is the reference velocity and acc is the acceleration. All three are parameters in Fig. 7.24.

It can also be exported as a *.lp file. For loading it from a GWL script, use the command `psLoadpowerProfile`, and choose the appropriate profile with the command `psPowerProfile`.

It is recommended to copy the .lp file in the same directory as the GWL script. If no file is mentioned, only the standard power profiles are available. ("IP Resist" and "None")

```
psLoadPowerProfiles "your_custom_profile_set.lp"
psPowerProfile "your_profile"
%MessageOut "Perfect Shape Profile"
```

Listing 7.26: Load own PowerProfile

8. Reference DeScribe

Many CAD applications can be used to create complex 3D models for the Photonic Professional (GT). This chapter explains how to prepare 3D models exported from a CAD application for direct laser writing.

8.1. 3D CAD software

The Photonic Professional (GT) is compatible with any CAD application exporting 3D models to the Standard Tessellation Language (STL) file format. DeScribe converts STL files to the General Writing Language (GWL) file format, which can then be loaded into NanoWrite for printing. The STL file format defines the closed surface geometry of a three dimensional solid object. During the export to STL, the original 3D solid model is approximated with a triangular surface tessellation (Fig. 8.1). Although many CAD applications can export to STL, there is significant variation in the quality of the surface model generated during the export. A good model without holes or self-intersections is essential for further processing. We have found the following commercial CAD applications to be capable of generating STL output of sufficient quality: AutoCAD, Autodesk Inventor, PTC Creo, and SolidWorks. Alternatively, STL files can be created with 123D Autodesk[1] which is available for download free-of-charge and comes with a collection of online video tutorials.[2] STL files of varying quality can also be downloaded from the Internet. Some common defects in STL files can be repaired with the open-source software Meshlab.[3]

8.2. DeScribe

DeScribe is the GWL editor developed by Nanoscribe for laser lithography applications. It contains syntax highlighting of GWL commandos, a debugger and pre-visualization of the structure. In addition, an import wizard reads STL files exported from your CAD application and produces a GWL output. DeScribe is installed on the Photonic Professional (GT) and can be run in parallel with NanoWrite. However, DeScribe does not depend on NanoWrite and can also be installed on other computers for offline preparation of 3D model data. In fact, for previewing bigger structures a well equipped office computer is highly recommended.

DeScribe installation on a notebook or computer is possible only if the following requirements are met:

- **Windows version**: DeScribe runs on Microsoft Windows 7, 8, 10 or Windows Vista with Service Pack 2 (or later).

- **Graphic Card and drivers**: WDDM 1.1, DirectX 10 compatible Hardware, DDI 10

[1] `http://www.123dapp.com/create`
[2] `http://www.youtube.com/user/123d`
[3] `http://meshlab.sourceforge.net/`

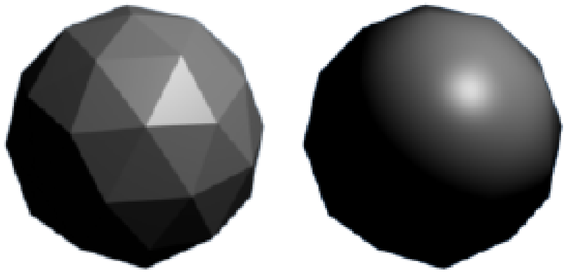


Figure 8.1.: On the left, a structure is represented with a small number of triangles. On the right, a smooth representation of the same figure.

To check the WDDM version, go to Start > Run, enter "dxdiag" to launch the Direct diagnostics tool, and switch to its Display tab. The WDDM version should be listed as the last item of the Drivers section on the top right. If you have WDDM 1.0 only, you will have to upgrade your graphics drivers to a newer version that is compatible with WDDM 1.1.

In some cases the system will report DirectX 11, although the hardware does only support older versions of DirectX. Check the DDI Version. It has to show at least Version 10.

- **Software**: .NET Framework 4.0: will be installed by DeScribe installer. Internet access is required if not present.
- Recommended: 32 GB of RAM as well as at least 4 GB of RAM on a dedicated graphics card

If you have any question please contact the Nanoscribe service team at service@nanoscribe.com

8.2.1. Working principle

DeScribe reads this data from the STL file and slices the object by intersecting the 3D solid surfaces with a series of parallel planes. For each plane, DeScribe computes the intersection contours of the surfaces with the plane and subsequently fills each obtained contour with a line hatching. Finally, the computed slicing contours as well as the hatched layers are saved in a filename_data.gwl file. A second file named after filename_job.gwl includes the data file and contains all relevant GWL parameters. Furthermore, DeScribe creates a compiled (=machine-readable) version of all required files in a gwlc subfolder.

Subsequently the filename_job.gwl file can be loaded in NanoWrite. If the latter can find the compiled files, the loading will be almost instantaneous. Otherwise, NanoWrite will start compiling. Note that DeScribe is much more efficient at compiling GWL files than nanowrite.

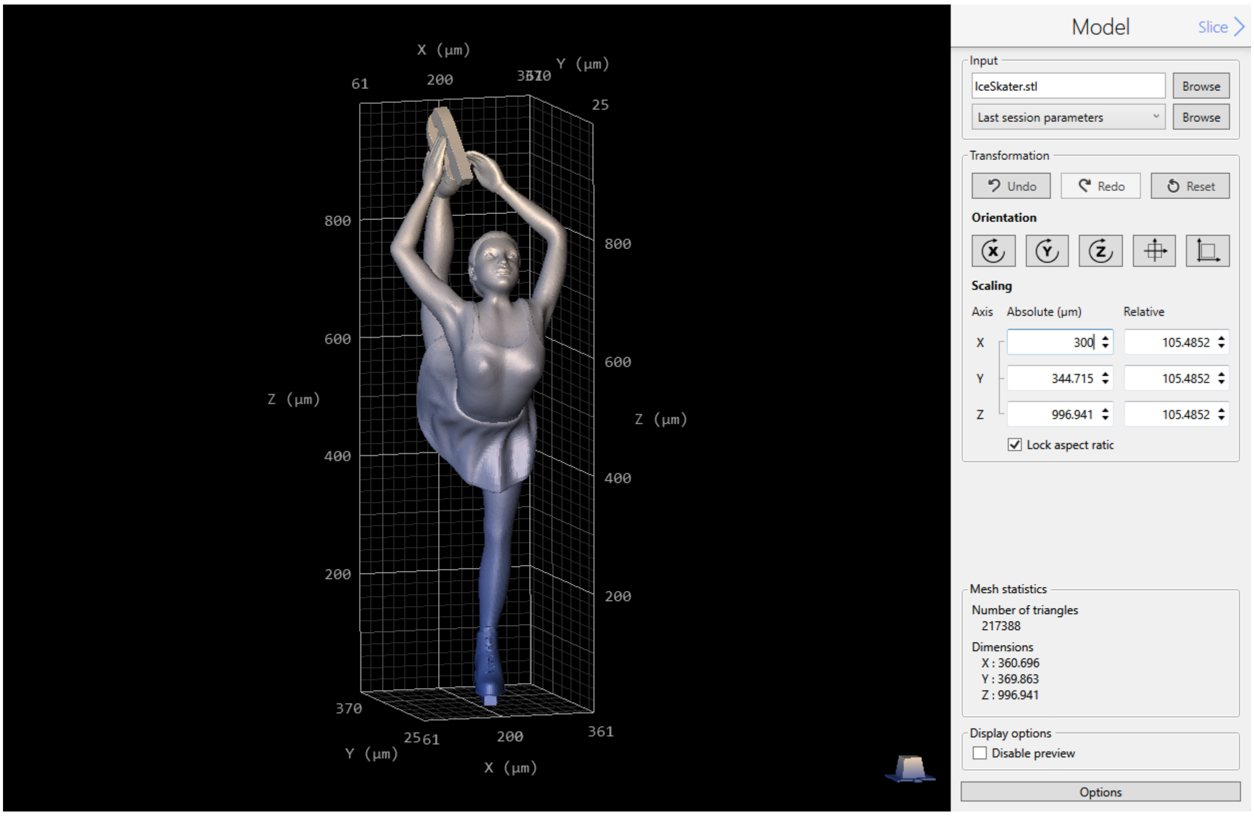


Figure 8.2.: The ice skater design can be visualized in DeScribe. The model can be resized and rotated after opening.

8.3. Using DeScribe

Loading data

Open DeScribe and load a STL file either by clicking File -> Open and choosing the appropriate STL file, or through a drag-and-drop of a STL file into the DeScribe window. Once the file is loaded, the import wizard automatically opens up.

Description of the design

The design considered is the figure of a dancing ice skater. Figure 8.2 shows the STL import wizard with the preview of the model opened with DeScribe. The wizard is divided into five steps, Model, Slice, Fill, Scaffold and Output, which are shown one at a time on the right hand side of the preview, as shown in Fig. 8.2 and 8.3. In the following, each step will be explained in detail.

Model

The first step *Model* allows to resize the dimensions, as well as change the orientation. The user can also load recipes in the **Input** box. All parameter from the import wizard are saved in the recipe. DeScribe comes with a set of predefined recipes. However, each time a STL object is imported a recipe will be saved with the job GWL file. This recipe can be loaded afterward. The set parameters from the last session are saved automatically and can be loaded from the drop down menu in **Input** as well.

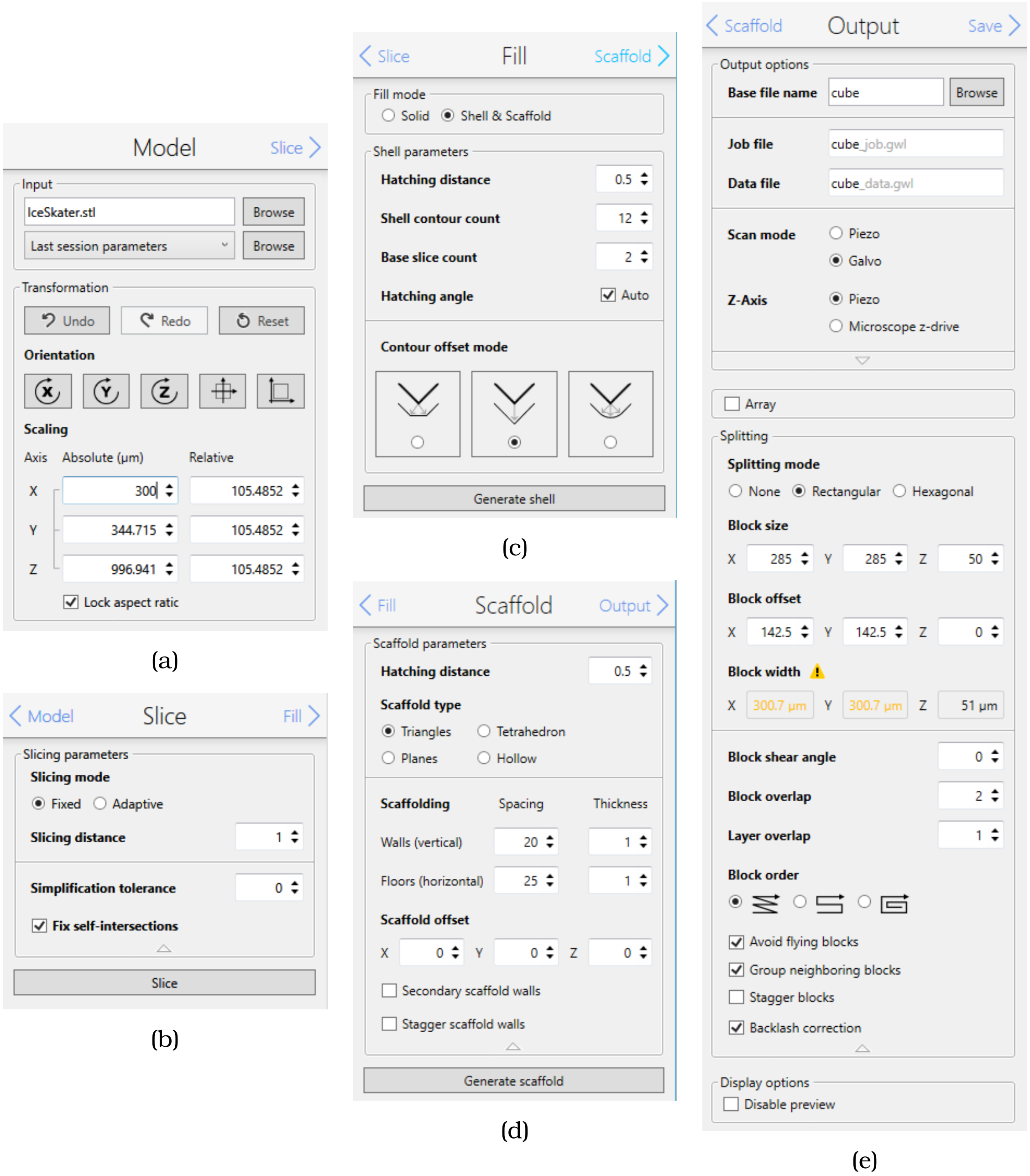


Figure 8.3.: DeScribe STL import wizard: Model, Slicing, Hatching and Output.

In the **Scaling** wrap box the final dimension of the design can be set. By checking the **Lock aspect ratio** box the proportions among the parts are kept constant; if not the individual side length along the respective axis are altered independently. Additionally, the structure can also be rotated around x, y and z-axis using the buttons found in the **Transformation** box. If the relative position of the design to the writing coordinate system is critical, the design can be also aligned to pre-existing markers on the substrate. For a detailed explanation of the procedure please refer to Chapter 7.7.

In the same box the user can choose which coordinates will be used. This depends on the chosen hardware: Galvo or Piezo. For `GalvoScanMode` all four quadrants with the origin in the center of the objective writing field are used. Therefore positive as well as negative coordinates are valid. In `PiezoScanMode` the first quadrant is selected, respectively only positive coordinates are valid, which corresponds to the addressable volume of the Piezo hardware. In both cases the coordinate system for the stage is untouched and negative stage displacements are always possible.

Below **Transformation**, two additional boxes are found: **Mesh statistics** and **Display options**. The number of triangles and the total size of the structure give an idea of how long it will take to divide it into layers. The more complex and bigger the structure is, the more time is needed for computation. Regarding the Display options, if you are working on a PC with a poor graphic card, it is advisable to check the **Disable preview box**. The graphical representation of the sliced structure is very demanding for the graphics card, and can lead to memory overflow. In such case, DeScribe is interrupted by the operating system and might unexpectedly crash.

Slice

The structure will be built by stacking layers, also called slices, on top of each other. The distance between the layers has to be adjusted according to the objective & resist combination in use and the degree of curvature of the surface. For the slicing mode, there are two options available:

1. **Fixed:** The distance between adjacent slices is constant and DeScribe computes the number of equidistant slices that still fits into the give height. This means that for a slicing distance of 1 µm and a 100.9 µm tall model, the last slice is at 100 µm.

2. **Adaptive:** Activating this option, DeScribe uses the slope at the current layer to determine the distance to the next slice. A steep slope will result in a higher slicing distance and a shallow slope will result in a smaller one. This is useful to adapt to the curvature of the design when surface roughness shall be reduced. The resulting slicing distance will be in the range specified by 'slicing distance' and 'minimum slicing distance'.

Slope Evaluation is meant for models which are not rotationally symmetric and steep as well as shallow local gradients are found within the same slice. Steep gradients are better approximated to a continues curve than shallow ones. Therefore, determining an intermediate gradient for each plane helps to minimize the staircase shape of the layer stack. With **Slope Evaluation** gradients around one plane are set. This computed gradient can be tuned between the **Average** of all the diverse gradients around the plane and the smallest gradient found (**Max**). Hence, setting **Slope Evaluation** to **Average** corresponds to a compromise between reducing the number of layers and approximating

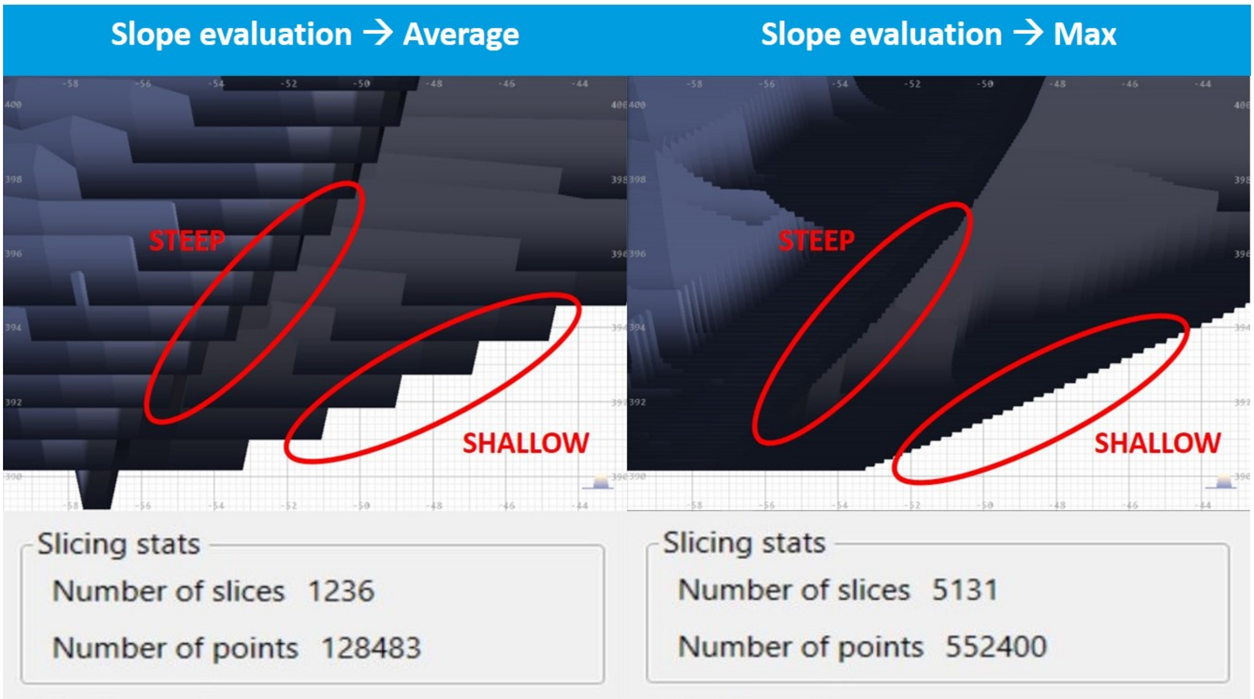


Figure 8.4.: Slope evaluation. On the left side Slope evaluation is set on Average. The staircase appearance is pronounced, but the number of layer is low. On the right Slope evaluation is set to Max. The surface becomes smoother because the slicing distance is small. Therefore, the number of layers increase accordingly. The red circles on both pictures mark the difference in surface roughness for different gradient.

the curve, whereas **Max** results in a smaller slicing distance and a smoother surface (Figure 8.4).

Surface normal smoothness: The slope is calculated using the normal vector at the surface of the model. 'Flat' surface normals use per-triangle normals while 'smooth' surface normals use pervertex normals which are the average of the (flat) normals of all triangles which meet at that vertex. Note that this option does not change the surface of the model! It only affects the normal vector and thus the slope value which then affects the slicing distances. It has a noticeable effect only for models with very large triangles (as compared to the slicing distance). For most models the differences can be neglected.

Fill

Fill is a defining step which separates mesoscale structures from high resolution printing. The differentiation starts with the **Fill mode** choice: **Solid** for high resolution, and **Shell&Scaffold** for mesoscale.

Selecting **Solid**, the core of the structure is filled with a dense set of adjacent lines. Their spacing is defined by the **Hatching distance**. The design can as well be smoothed by adding one or more contours and if more than one contour is used, the distance between these shells can be set with **Contour distance**.

In order to increase the mechanical stability of the structure or to decrease the writing time a **Hatching angle** and a **Hatching angle offset** can be defined. The **Hatching angle** sets the angle between the hatching line direction and the x-axis. The **Hatching angle offset** sets an angle offset between the orientation of one layer of lines and its consecutive.

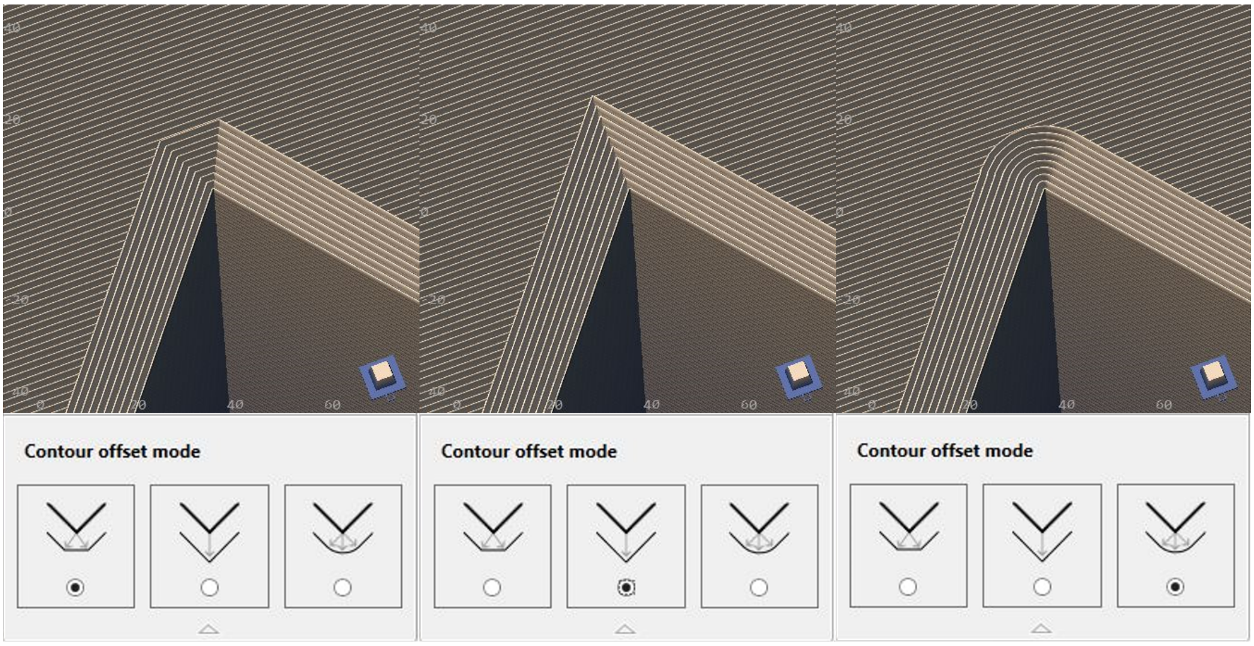


Figure 8.5.: Contour Offset Mode. From left to right the pictures show how Beveled, Sharp and Round mode shape the inner verteces of the shells.

For example, to increase the stability of a thin wall it might be helpful to set the hatching line orientation along the axis of the short side of the wall. However, this setting will increase the number of lines per layer, hence the writing time of the structure. In such case, the user can check the **Auto** box and DeScribe will choose the **Hatching angle** as well as the **Hatching angle offset** which results in a shorter printing time.

Finally, the **Contour offset mode** specifies how the vertices of inner shells are going to be reproduced: **Beveled**, **Sharp** or **Round** (Figure 8.5). **Sharp** takes the shortest time to compute and the resulting file size is the shortest, as less points are required to define the structure. **round** is set as default (Loading DeScribe for the first time). However, **sharp** is saved for all **Shell&Scaffold** recipes.

Notice that the Fill window changes slightly by choosing Shell&Scaffold and there will be two different settings available. **Shell contour distance** is equivalent to **Contour distance**. However, its purpose is different. The shells are necessary to prevent the developer from creeping into the structure during the development process. Therefore, to have a hermetic shell the **Shell contour count** should not be chosen lower than 12.

As well, base layers improve the stability of this resist-containing vessel by reducing the deformation due to shrinkage. It was observed that 5 layers strengthens the structure and increase adherence to the substrate. All these features can be visualized and understood better by using the **Clipping option**. By sliding the knob from right to left the structure preview will be cut through by a plane perpendicular to the checked axis. Thereby, the features which are masked by the shell will be visible.

Scaffold

This step is available only if **Shell&Scaffold** is checked and defines how the elementary units of the scaffold are going to be arranged inside the shell.

Hatching distance defines the pitch of the lines used to build floors along the z axis. Those floors are going to be subdivided into elementary cells which can be either triangular prism (**Triangle**), tetrahedron **Tetrahedron** or rectangular cuboid (**Planes**). Their dimensions are set by the spacing of the vertical (**Walls**) and horizontal planes (**Floors**), and their structure can be reinforced by increasing the number of planes per wall/floor (**Thickness**). The distance between the planes is determined by the **Hatching distance** as well. Finally, the scaffold mesh can be shifted using an offset, increase its density by adding secondary walls and enhance its complexity shifting each plane of cells in a staggered fashion.

In the Display options the user can decide to make visible only the scaffold, the outer shell or both of them simultaneously.

Output

The Output menu finalizes the conversion process. Two files and one folder are generated: filename_data.gwl, filename_job.gwl and filename_file folder. The filename_job.gwl is the one file which can finally be loaded into NanoWrite for laser writing. Nevertheless, to successfully print the structure, both the two files and the folder should not change directory. If it is necessary to have the files saved in different folders please refer to 7.1.7.

In this menu, fundamental parameters are set:

- **Scan mode and Z axis**: Defines which units of the system are going to be used to write. For mesoscale structures it is advised to use the combination of Galvo and Microscope z-drive.
- **Exposure**: **Constant** allows only one value for laser power and one for scan speed throughout all the structure, whereas **Different** gives the possibility to set different variables for shell and scaffold. Their values can be modified at any time by opening the filename_data.gwl file, at the top where the corresponding variables are declared.
- **Z-Direction**: It specifies if the structure will be written bottom to top or top to bottom. Bottom to top is used only in oil immersion configuration in which the resist sits on top of the substrate. Top to bottom is required if writing in standard configuration and/or inside microfluidic devices. This writing direction avoids scattering in already polymerized layers, as it is the case for an oil immersion setup. Further this configuration allows to write structures taller than 40 µm in standard configuration.
- **Hatch lines**: You can choose between back-and-forth hatching and single direction hatching. The single direction hatching is designed mainly for the `PiezoScanMode`. When writing a straight line, there is a small shift in the perpendicular axis. As a result, writing all lines in the same direction keeps this offset the same for all lines. However, in back and forth hatching this offset is alternatively changing in direction, leading to pairing of lines. In `GalvoScanMode` this effect is not observed and back-and-forth hatching should be favored.
- **Array**: This feature was developed for structures smaller than the printing field. Figure 8.6 helps to understand its possible output. To print an array and split the structure, the *Advanced STL processing* workflow *Parameter Sweep* has to be used. This feature takes the working distance and structure height into consideration to set the structures in a distance so that the objective lens won't be damaged.

Once all parameters are set, the two GWL output files are saved in the same directory as the STL input file along with a recipe file.

File Structure

The Job file contains the standard header corresponding either to `GalvoScanMode` or to `PiezoScanMode`. These standard headers are detailed in Section B.12.

The data file contains the GWL data of the structure. The parameters used in the slicing process are stored as comments.

Multiple jobs can be combined in a header GWL file with a `include` command. The stage needs to be translated, using `movestageX` for the x-axis, to avoid overriding of the previous written structure:

```
include cone_job.gwl

movestageX 200
include cube_job.gwl
```

Listing 8.1: Combine jobs

8.4. The workflow from the CAD design to the print

The complete workflow for a quick start with your first structure is the following:

1. Design a structure with any CAD software and export it as a STL-file.
2. Open your STL structure in DeScribe (Fig. 8.2).
3. Follow the conversion process, with the Model, Slicing, Hatching and Output steps (Fig. 8.3). The created GWL-job file is opened in DeScribe.
4. You may also preview the structure (press F5).
5. Check all header settings, particularly laser exposure and scan speed parameters.
6. Remember that achieving the optimal structure usually requires some trial and error testing. As a first step, you can plan a dose test where the laser power is varied for the same structure at different locations. It is important to keep in mind, that if the power compensation is activated, the laser power is defined inside the generated GWL-files. Presetting a `LaserPower` for the whole structure in the main GWL-file is therefore ineffective, since the value will be systematically overwritten. If you wish to perform a dose test, you may vary the `PowerScaling` value instead. Indeed, the product of `PowerScaling` and `LaserPower` is the effective laser power. In your main file, the part following the header looks like this for the dose test (refer to figure 8.7 for a preview):

```
PowerScaling 0.6
include cone_data.gwl

AddXOffset 20
PowerScaling 0.8
include cone_data.gwl
```

```

AddXOffset 20
PowerScaling 1
include cone_data.gwl
```

Listing 8.2: Simple dose test

7. Start NanoWrite and follow the experimental workflow as detailed in chapter 5

You are now ready to proceed in NanoWrite. Note that the writing parameters, such as the speed or laser power may not yet be optimal in your experimental conditions. The experimental work starts here.

As a first step, a dose test as described in section 7.1 is highly relevant. Using `Include`, `XOffset`, `YOffset`, `MoveStageX`, `MoveStageY`, `LaserPower` or `PowerScaling` commands, you will be able to create your first successful job. However, in order to really consider all the details, you will need to read through chapter 7. Please, also see the command reference B.

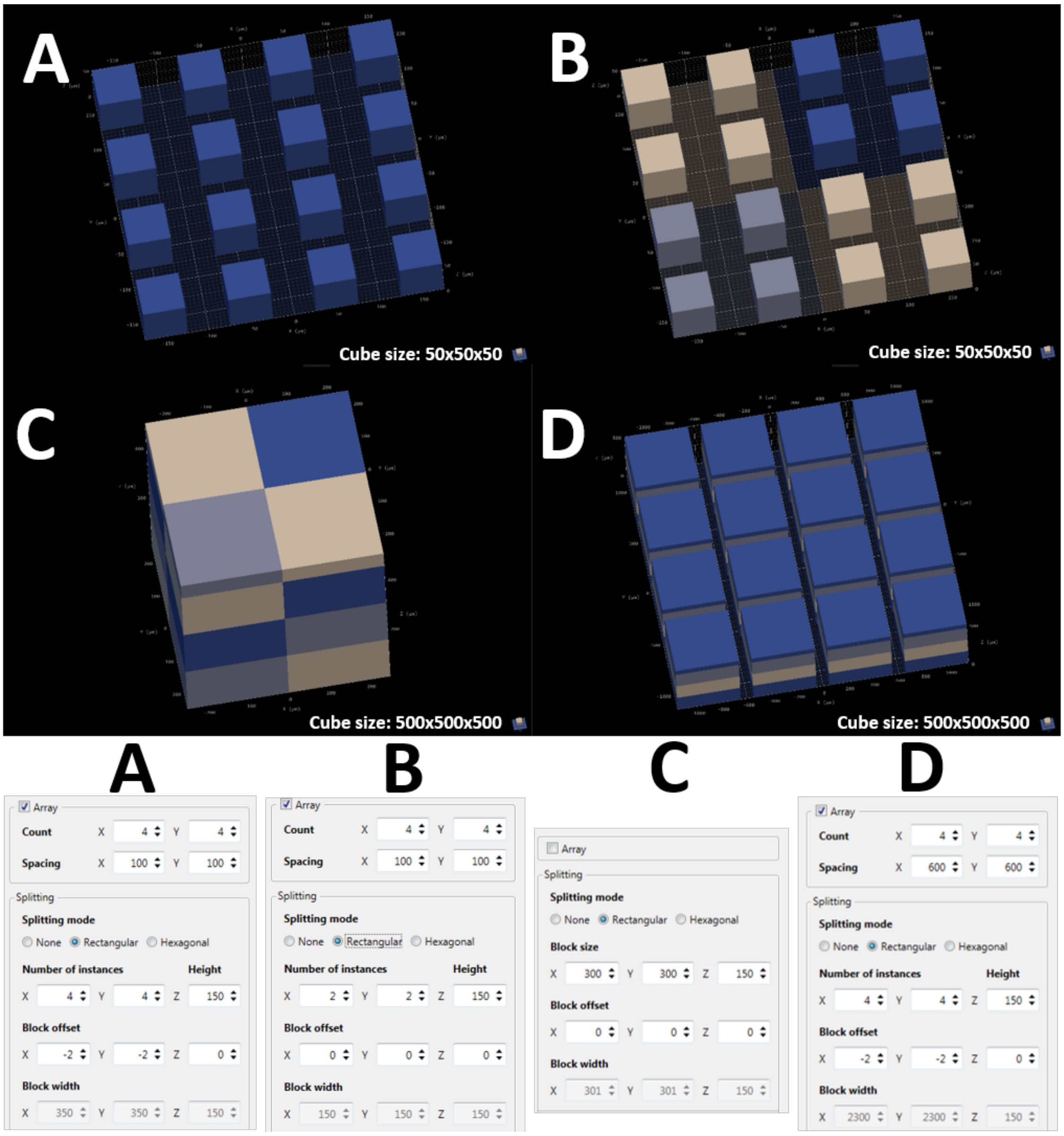


Figure 8.6.: Array feature: A) An array of 4x4 cubes with a size of 50x50x50 µm^3 was created with a distance of 100 µm (center to center). The splitting was set in such a way that all cubes lay within one printing block. B) The splitting mode was changed, so that 2x2 cubes lay within one printing block. C) The array function is disabled, hence, only one cube is printed. However, the cube size was increased to 500x500x500 µm^3. As the block size was set to 300x300x150 µm^3, the cube is split and will be printed as many blocks. The block shear angle is set to 0° in this case to better illustrate the splitting and array functionality of DeScribe. D) The 500x500x500 µm^3 cube was set in an array of 4x4 with a distance of 600 µm. The combination of array and splitting is only possible in the *Advanced STL processing* workflow *Parameter Sweep* which takes the working distance and feature size into consideration to set the structures in a save distance to each other to protect the object lens. To indicate this circumstance the blocks are just cut in layers. (If the array function is disabled, the block size can be defined directly. If the array function is enabled, the block width is defined by the number of instances or repetition of the same object which are included in the block, their size and distance to each other. A warning appears if the structure size exceeds the scan field of the objective. In the array mode the block offset refer to an offset of block instances.)

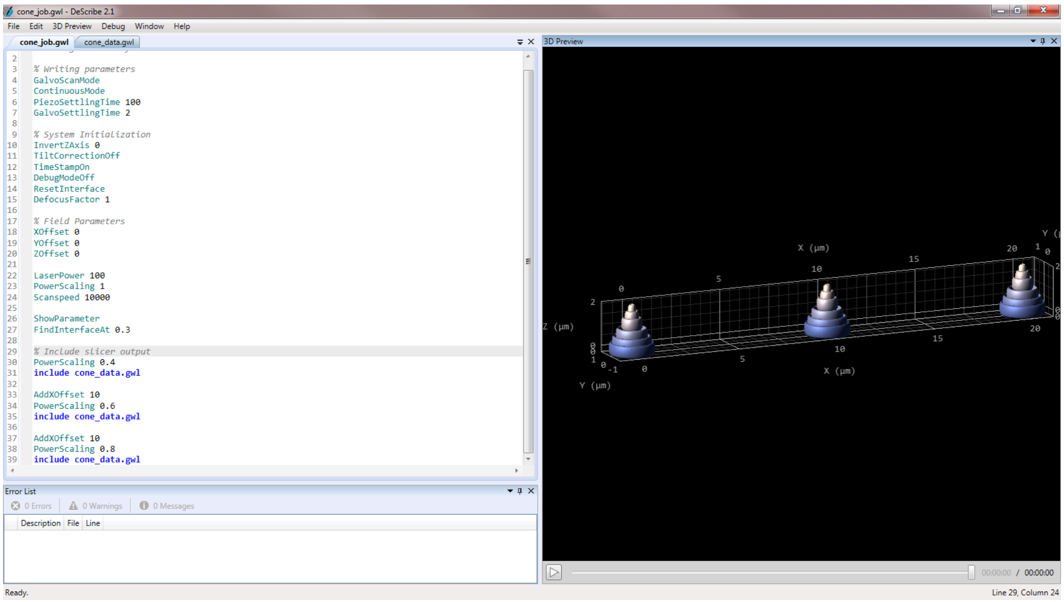


Figure 8.7.: DeScribe screenshot of a simple dose test. In the main file, there is no use to set any `LaserPower` value since it would be then overwritten. Instead, we set different `PowerScaling` values.

A. Care and Maintenance

Preliminary remarks

The Photonic Professional (GT) itself is almost maintenance free. But as with every open system that contains moving parts and mirrors, cleaning from time to time may become necessary. Most parts of the system, however, should only be cleaned if a bad performance of the system has been found. For the parts that get in touch with chemicals this doesn't apply. **Stage and objective lenses need to be cleaned regularly and felt rings have to be replaced as needed.**

In general, we do recommend wearing gloves for the cleaning process. Please make sure that the laser emission is off during the cleaning process. Furthermore, take care not to remove any cables, because some of them are definitely not hot-plugable. For safety reasons, we recommend shutting down the whole system.

If you are in doubt please contact service@nanoscribe.com or call us at +49 721 981 980 4.

A.1. Cleaning the system

The exterior of the system should be cleaned to maintain a clean working environment. Please use a slightly damp tissue and avoid cleaning agents. You may also use isopropanol for all surfaces. Note, however, that the breadboard and other stainless steel surfaces may get stained if using isopropanol. Instead use warm water and thoroughly dry it with a dry lint-free cloth immediately afterward.

Details about the cleaning procedure for specific parts of the system are listed below.

A.1.1. Cleaning of objective lenses

The objectives are the only part of the system that needs regular cleaning. Depending on your usage and environment you will have to adapt the intervals. There are different viable ways to clean an objective. One is described by Carl Zeiss "the clean microscope" `https://www.zeiss.de/mikroskopie/produkte/microscope-components/objektive.html#downloads`.

The recommended method by Nanoscribe will be explained in detail below. Note that special requirements for the different objective lenses are listed separately. Please read all the information before starting the cleaning process.

Note that the metallic part of the objective is rather sturdy and easy to clean. The small center glass lens however is fragile, easily scratched and damaged. This is the part that is crucial for the performance of the system and may under no circumstances be touched by any hard or scratching material (any quality cleanroom paper or tissues do scratch and therefore unsuitable!).

Objective removal

Warning!
Make sure nobody is using the touch panel of the microscope while you are removing or reinserting the objectives into the revolver.

1. Stop any writing process in NanoWrite.
2. Press 'exchange holder' in NanoWrite (Fig. 4.1) and make sure the objective is in the lowermost position (message "lower z-drive position reached" should be visible on the touch panel).
3. Stop the laser emission by pushing the *laser emission* button (Fig. 2.10). Leave the laser key in the on position.
4. Select the turret position which is two steps further (clockwise) than the objective you want to clean. E.g. if you want to clean the objective in position 5, select position 1 (the nosepiece contains 6 positions). This way the desired objective will be facing sidewards to the right of the microscope. Try to remember roughly the angle the objective is facing towards you (you will need to put it back in with the same angle to screw it back into the turret).
5. Find the objective casing, open it and keep the black cover for the objective nosepiece in reach.
6. Unscrew the objective lens counter-clockwise from the side of the microscope (below the stage). Use your left hand to make sure you will not drop the objective and the right hand to unscrew it. As soon as it comes lose, tilt it forward into the center of the revolver and then move it out backwards. Avoid any mechanical contact to the objective top (front lens).
7. Mount the objective into the black part of the casing and put it aside. Place the black cover onto the turret (only pushing, no screwing involved).

Cleaning process

We recommend to only use highly-clean isopropanol (purity $\geq$ 99.9 %) to avoid residues on the lens surface. DO NOT use any strong solvents, acids, acetone, etc. as they might damage the lens or its cementation (fixation).

1. Remove the suction ring (made of felt, kind of a sponge).
2. Wipe around the front lens with a clean tissue to remove most of the resist/oil. Avoid the glass lens itself. See fig. A.2 (a).
3. Use a cloth/paper to wrap the objective without touching the lens. This is to avoid any liquid entering the inner volume of the objective, see fig A.2 (b).
4. Hold the objective in about a 45° angle and use a pipette to drop individual drops onto the center glass lens. These drops should be captured by the paper wrapping.

Figure A.1.: Objective taken out and fixed onto black mount.

5. Blow the top of the objective until its dry. It's best to use Nitrogen or a dust blower (do not use air cans or pressurized air).

6. Repeat the last two steps until there is no more residue visible and then repeat two more times.

7. If this does not clean the lens sufficiently you might have to use lens paper as described by Carl Zeiss in "the clean microscope" `https://www.zeiss.de/mikroskopie/produkte/microscope-components/objektive.html#downloads`.

8. If you are confident the lens is clean, replace the suction ring. Make sure to use a new ring if you got the impression the ring cannot take up much more chemicals.

9. Remove the black cover from the same objective turret position, where you placed it before.

Remounting the objective

1. Remove the objective from the black casing.

2. Move it slowly inside the area between the revolver and the bottom of the stage with the glass lens facing forward (inside). Stay as low as possible. Once the back of the lens (the part with the threads) is over the hole, tilt the lens upwards in its correct position. Avoid any mechanical contact to the objective top (front lens).

3. Screw the objective clock-wise: Again use one hand only for safety and the other hand to actually rotate the objective. Note that you will need a slight angle to screw the objective in. It is tight enough once the revolver starts rotating when you screw the objective in.

4. Choose the objective on the touch panel.

(a) (b)

Figure A.2.: How to use tissues for the cleaning of the objectives: (a) Cleaning around the lens by rotating the lens and holding the tissue close to the center without touching it. (b) Wrapping the objective in a tissue order to collect the drops of iso-propanol.

5. Turn the emission of the laser back on and continue your work. A `recalibrate` procedure is advised.

Measures with respect to objectives

Note!
The immersion objective (oil & resist) have a suction ring around them to avoid any liquid running down the objective and making its way into the nosepiece. Once this kind of sponge is full, it has to be replaced with a new one immediately!

Oil immersion objectives Since the immersion oil collects the dust and becomes bad (greenish color) after a while, regular cleaning intervals are recommended. Make sure not to apply a lot of immersion oil onto your substrates in the first place. Otherwise, the suction ring becomes full quickly and you run the risk of contaminating the nosepiece with oil. In such a case the microscope needs to be sent in for a costly cleaning by the manufacturer.

DiLL objectives These objective are used with a resist/resin as an immersion medium. These materials may become hard (polymerize, etc.) and in effect are not removable from the lens surface anymore. There is a real risk of completely damaging the objective. Therefore take very good care not to expose the resist/resin on the front lens. DiLL objectives should only be used with IP-Dip resist and handled in a yellow-light environment.

We recommend to clean DiLL objectives whenever you think they will not be used for a while or if the next user of the system uses a different objective. Also, if by accident, you have mixed immersion oil with resist, e.g., by approaching with the wrong objective, you should immediately clean the objective and use a new sample. Additionally, we recommend to store the DiLL objective in a dark, light-shielded place if it will not be used for an extended period of time.

A.1.2. Cleaning of stage and objective revolver

Caution!
Do not clean the stage while it is moving, especially not during the calibration procedure. This movement cannot be interrupted and can lead to severe injuries.

Usually stage and objective revolver only collect some dust and a little bit of immersion oil, which touches the stage during the insertion of the sample holder. However if you use too much immersion oil, you might find that the oil starts to spread and a lot of cleaning becomes necessary. In such cases, please inform all the users to use less immersion oil and clean the stage immediately. Also note that the oil becomes greenish after a while - please don't wait that long before you clean it off!

Both stage and objective revolver can be cleaned with a lint-free isopropanol wetted cloth. To be able to reach as much surface as possible, you can use either the joystick or Märzhäuser (switchboard) software to move the stage to its extreme positions. Please make sure the objective is lowered and NanoWrite is off before doing so.

A.1.3. Cleaning optical surfaces

As a general measure for cleaning optical surfaces such as beam splitters, mirrors or filters, it is necessary to:

- Work with gloves to avoid any finger contact to the active surface
- Blow off dust first using a blow-ball or nitrogen
- Only use isopropanol and/or distilled water as a second approach
- Eventually remove dust and stains with a clean, soft and absorbing tissue
- Dry the surface afterwards with a blow-ball or nitrogen

Depending on the working environment, either in a cleanroom or not, and after extended operation time, dust particles and chemicals (immersion oil, photoresist) may drop on the different optical surfaces along the light path in the microscope. This may result in unexpected scattering of the incoming laser beam, instability of the autofocus procedure or degraded image from the live view camera. We recommend, once a year, to carefully check and if necessary perform the cleaning of all relevant optical surfaces in the microscope.

For further detailed information on cleaning of optical surfaces, we recommend to consult the Zeiss website: `http://Zeiss-campus.magnet.fsu.edu/articles/basics/care.html`.

Cleaning the beam-splitter in the reflector module

1. Make sure the nosepiece holding the objective revolver reached its load position, then shut down the microscope. On the right side of the microscope as seen in Figure A.3, rotate the locking lever (1) downwards and open the black cover (2) upwards. Now you can carefully pull the reflector turret (3) out of the microscope and place it on a suitable surface.

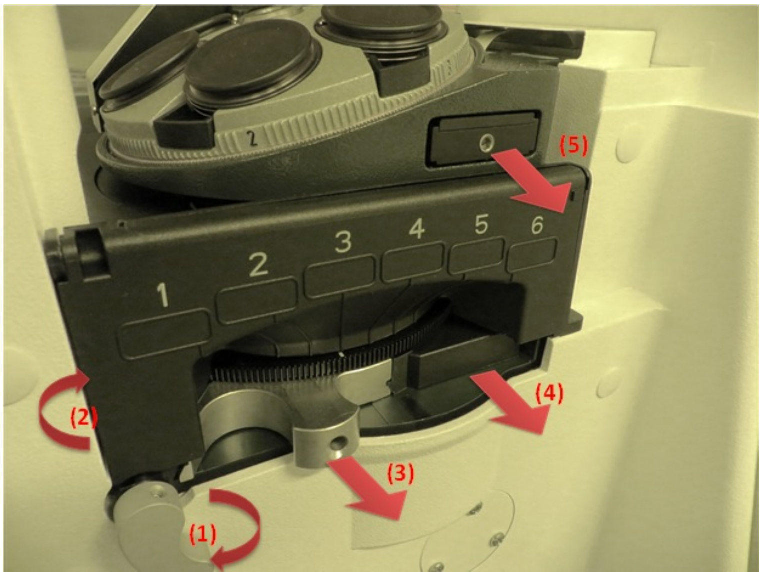


Figure A.3.: Side view of the microscope. (1) locking lever, (2) black plastic cover, (3) reflector turret handle (4) green filter holder (5) beam-combiner

2. The beam reflector module is located on the position 3 of the reflector turret, see Figure A.4. In most cases dust can be blown off without further disassembly. If not, remove the reflector module by slightly lifting the upper string clips and then tilting the module outwards to take it out. After it comes free you should be able to properly inspect the optical surface.

3. The beam splitter is the angled optical surface, of the reflector module, see Figure A.4. Its dielectric reflective surface points downwards and should not be affected by dust or chemicals, so focus on its upper side. It is not recommend to disassemble the reflector module. In case of the presence of dust carefully use a blow-ball, or a nitrogen gun. After inspection reload the reflector module in the turret and insert it back into the microscope. Do not forget to pull down the black cover and rotate the locking lever upwards.

Cleaning the Definite Focus beam combiner

The beam combiner for the DefiniteFocus module is located above the reflector turret and beneath the objective revolver (Figure A.3 (5)).

1. In order to make it accessible you have to move the nosepiece upwards so that the socket with the beam combiner module is freed from the black cover (Figure A.3 (2)). Note that for manually moving the nosepiece upwards, the microscope has to be on.

Caution!
Make sure that you removed the sample holder from the piezo stand. Otherwise this could lead to a crash of the objective into a substrate, and could damage the lens.

Use the respective Allen key to unlock the beam combiner module by rotating key screw by a little less than 90°. Then carefully slide it out of its socket.

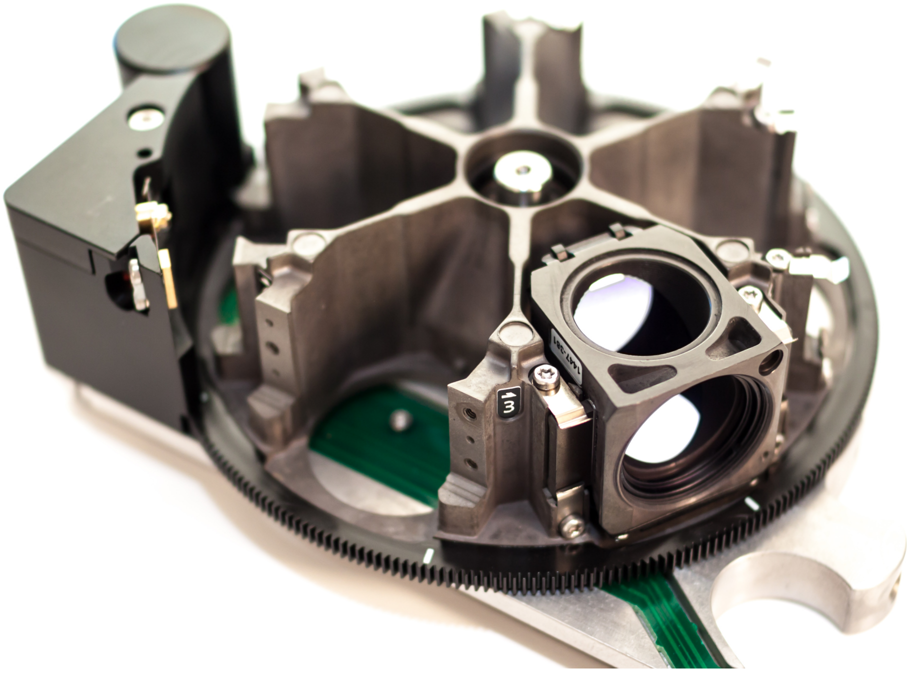


Figure A.4.: View of the turret with beam reflector (splitter) module.

2. The beam combiner, Figure A.5, has a fragile surface. Please do handle the module very carefully and never touch the glass surface with your fingers. If dust is deposited on the surface you might blow it off with a blow-ball or a nitrogen gun. If possible avoid cleaning it with isopropanol if it is not necessary. However, if stains from chemicals happened to be present, clean it carefully with isopropanol and a clean and soft tissue. Dry it with air or nitrogen. When clean insert the beam combiner back in its socket, lock it with the Allen key and press Load Position on the touch panel so that the nosepiece moves to the lower z-limit.

Cleaning the optical elements in the light path to the camera

The green filter is located right next to the reflector turret handle (Figure A.3 (4), A.6).

1. Pull it out and clean the surface of the filter with isopropanol and a clean and soft tissue.
2. Dry it with a blow-ball or nitrogen.

The blue Live Cam is located on the left hand side of the microscope (see figure A.7). Cleaning can be done as follows:

1. Shut down the microscope and turn off its power supply module. Turn the computer off (the camera is not hot-plugable!).
2. Unplug the FireWire cable from the Camera.

Note!
Unplugging the FireWire cable is to be done ONLY once the computer and microscope are off.

Figure A.5.: View of the beam combiner.

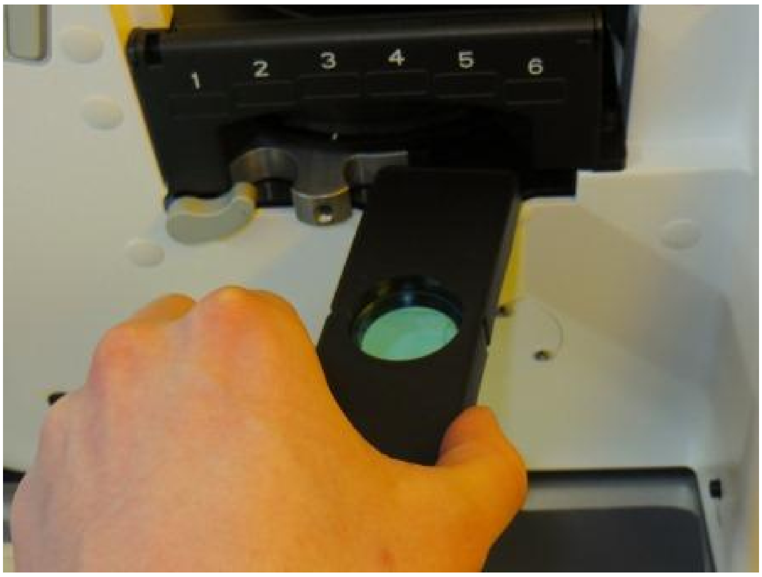


Figure A.6.: Removing the green filter which is located next to the reflector turret handle.

3. Loosen the camera from its mount by rotating the silver wheel. The first optical surface is the protection glass, that you may clean according to the standard procedure described before.

4. You can screw off the camera adapter (gray connection module), and clean the lens of this adapter.

5. Screw the gray connection module back on, screw the camera back to the stand and plug-in the FireWire cable.

6. Turn on the computer, and then the microscope power supply and the microscope.

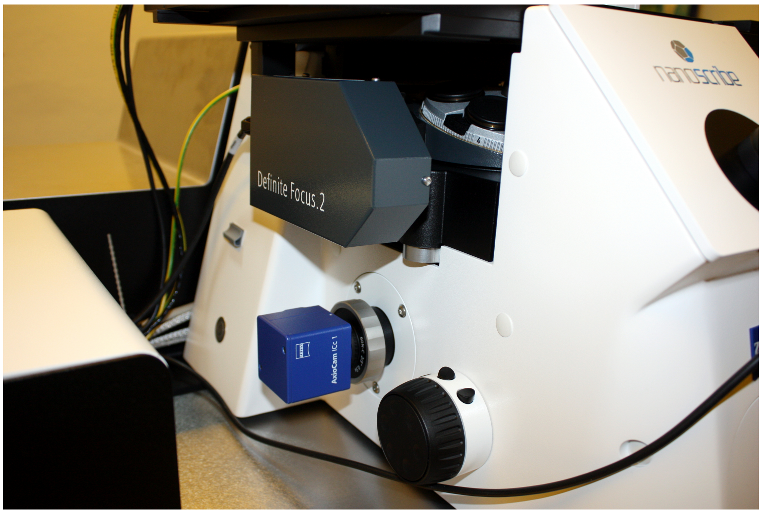


Figure A.7.: View of the live camera (blue cube) next to the Definitive Focus on the left hand side of the microscope.

A.1.4. Cleaning the Optical Cabinet

Caution!
You should not open the optical cabinet without consulting Nanoscribe service department (service@nanoscribe.com, phone: +49 721 981 980 4). There is a laser of Class 3B in there that is highly dangerous. Keep the optical cabinet closed for safety reasons and to avoid dust deposition on the optical parts.

The exterior of the optical cabinet should be cleaned with a slightly damp cloth.

A.1.5. Cleaning of the Keyboard, Mouse, Monitor, Webcam

We recommend to follow strict standards of cleanliness when working with our system, in order to promote quality and safety. Thus we highly recommend to use gloves for all work with chemicals and to take them off, before using the computer. This is for the safety of your health and that of your colleagues. Other than that, follow standard cleaning procedures for the input devices such as mouse and keyboard.

A.1.6. Maintenance intervals

As mentioned in the preliminary remarks, it is not recommend to clean any components of the system unless necessary. Such a condition is met, e.g., if one objective has a mixture of oil and resist on top. This might happen if you accidentally approach the wrong sample. Another reason would be that there is too much dirt in the optical path and therefore visible dirt in the live view becomes disturbing. We can thus merely offer guidelines on a schedule. The actual timing of the cleaning has to be decided as needed. Table A.1 gives a rough estimation on cleaning intervals in weeks for cleanroom and standard room installations respectively.

Table A.1.: Cleaning frequency (in weeks)

	cleanroom	non-cleanroom
Oil immersion objectives	4	2
DiLL objectives	2	1
Stage and revolver	12	6
Optical path in microscope	52	52
Outer surfaces	52	52
Mouse, Keyboard, Screen	52	52

B. GWL Command Reference

This chapter gives a complete list of all GWL commands. In addition, DeScribe has a build-in interactive help menu which pops up with the keyboard combination *Ctrl+Space* (Figure B.1). The more letters are typed in before *Ctrl+Space* is pressed, the shorter the reference list.

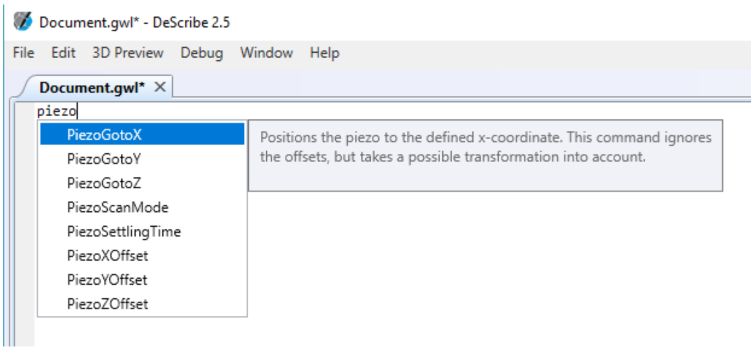


Figure B.1.: Interactive help menu of DeScribe.

B.1. Writing mode parameters

`PiezoScanMode`
`StageScanMode`
`GalvoScanMode`

In `PiezoScanMode` all coordinates are addressed via the piezo movement in x-, y- and z-directions. Please note that in this mode all coordinates are relative to the current position of the stage on the sample with the origin at the corner of the piezo. Switching to this mode the Galvo will be move to (0,0).

In `StageScanMode` the x- and y-coordinates are addressed via a stage movement whereas the z-coordinate is still addressed via the piezo. Please note that in this mode all coordinates are absolute with the origin at the sample center.

In `GalvoScanMode` all coordinates are relative to the current position of the stage *and* the piezo. In this mode the x- and y-coordinates are addressed by the galvo scan head whereas the z-coordinate is still addressed asynchronously via the piezo, i.e., layer by layer writing. Switching to `GalvoScanMode` the piezo will be set to (0,0,0).

`PulsedMode`

In this mode each programmed point is addressed individually. The exposure is paused between the points. This waiting time is defined by `SettlingTime`. The exposure time for each point is defined via `ExposureTime`. Please note that `ScanSpeed` and `UpdateRate` are not relevant for this writing mode.

`ContinuousMode`

In this mode programmed line segments are exposed continuously. The exposure is paused after each `Write` command. The writing speed is set via `ScanSpeed`.

`LogMode`

This mode is exactly like the `ContinuousMode` for the piezo. However, additionally one log-file per line segment (`Write` command) is created consisting of the following 8 columns: programmed coordinates [µm]: x, y, z; laser power [%]; effective piezo coordinates [µm]: x, y, z; diode voltage [V]. Please note that it is neither available in `StageScanMode`, nor in `GalvoScanMode`.

`ConnectPointsOn`
`ConnectPointsOff`

`ConnectPointsOn` interpolates additional points between the programmed coordinates in a linear fashion. The distance of the interpolated points is specified via `PointDistance`. These points are then sent to the piezo with the given `UpdateRate`.

For `ConnectPointsOff` no interpolation between the programmed coordinates is carried out. Therefore, only the coded points are addressed by the piezo.

This feature is applicable in `PiezoScanMode` during `ContinuousMode` as well as in `PulsedMode` and in `GalvoScanMode` during `PulsedMode` only.

B.2. Writing parameters

`PowerScaling` *value* — Default: *1.0*

Sets the factor for the power scale on which `LaserPower` operates. `PowerScaling` 1.0 is the default setting for which `LaserPower 0-100` operates between 0 mW and the reference power at the objective aperture, which is 20 mW or 50 mW for the GT. Higher laser powers than this can be reached by increasing `PowerScaling`.
For example to achieve 150% of the calibrated laser power you may use `PowerScaling` 1.5 and `LaserPower 100` or `PowerScaling` 2 and `LaserPower` 75. The actual setting for the AOM is derived by the product `PowerScaling` x `LaserPower`.

`LaserPower` *0-100* — Unit: *%*

Sets the laser power in percent on the current power scale.

`PointDistance` *distance* — Unit: *nm*

Sets the distance between the interpolated points when `ConnectPointsOn`.

This feature is applicable in `PiezoScanMode` during `ContinuousMode` as well as in `PulsedMode` and in `GalvoScanMode` during `PulsedMode` only. Watch out, the unit is nanometer! Useful values are mostly between 10-200 nm.

`UpdateRate` *rate* — Unit: *Hz* — Default: *1000*

Sets the rate at which the programmed or interpolated points are sent to the piezo.
This command is only valid in `PiezoScanMode`!

ScanSpeed *velocity* Unit: *µm/s* Default: *varies*

Sets the writing velocity in µm/s. This command applies for all ScanModes!
Please note that UpdateRate and ScanSpeed inter-depend. Changing the UpdateRate changes the ScanSpeed accordingly and vice versa.

ExposureTime *time* Unit: *ms* Default: *50*

Sets the exposure time in PulsedMode for each programmed or interpolated coordinate.

SettlingTime
PiezoSettlingTime
GalvoSettlingTime *time* Unit: *ms*

Sets the wait time between two line segments (in ContinuousMode) or between two points (in PulsedMode).

PiezoSettlingTime is necessary for the piezo to travel from the last writing position to the next writing position. Usually settling times between 100 ms and 500 ms are recommended depending on the travel distances between two lines or points.

GalvoSettlingTime is necessary for the galvo to settle down at the commanded position, i.e. for any oscillations caused by the previous movement to damp down. Usually, times between 1 ms and 3 ms are recommended. This command is only relevant in PulsedMode.

LineStartMode *1, 2* Default: *1*

Sets the waiting time prior to a line or a point exposure. LineStartMode 1 waits for the set SettlingTime prior to exposure. LineStartMode 2 does not wait for SettlingTime, instead the exposure takes place immediately.

This feature is not applicable for GalvoScanMode.

LineNumber # Default: *1*

Sets the number of lines that are written for one programmed line. This command is useful to quickly program gratings or for increasing the line thickness by writing multiple adjacent lines.

LineDistance *distance* Unit: *nm*

Sets the pitch of the lines added via LineNumber. Watch out, the unit is nanometer!

PolyLineMode *0, 1, 2* Default: *2*

Specifies whether the added lines generated via LineNumber are added

- 0 to one side of the programmed line
- 1 to the other side of the programmed line
- 2 on both sides (alternating) with the programmed line coordinates in the center i.e. with even LineNumber the programmed line itself will not be written, but instead lines on both sides of it.

PowerValues
PowerValuesOn
PowerValuesOff

Defines whether the 4th column of the coordinates will be used for defining the LaserPower.

`MeanderOn`
`MeanderOff`

`MeanderOn` uses alternating writing directions for the lines added via `LineNumber`.
`MeanderOff` uses equal writing directions for all lines added via `LineNumber`.

`Wait` *time* Unit: *s*

Waits the specified time in seconds before the writing procedure continues. A respective message will be issued in the message log.

`Write`

This command sets the end of a trajectory defined by a prior set of coordinates.

B.3. Positioning commands

`XOffset`
`YOffset`
`ZOffset` *distance* Unit: *µm*

Sets an offset for all programmed x|y|z-coordinates. The offset is added to each x|y|z-coordinate of each programmed or interpolated point. This command is useful to shift repeated structures in the writing area.

`PiezoXOffset`
`PiezoYOffset`
`PiezoZOffset` *distance* Unit: *µm*

Sets a piezo offset for all programmed x|y|z-coordinates. The offset is added to each x|y|z-coordinate of each programmed point. This command is only applicable in `PiezoScanMode`.

`StageXOffset`
`StageYOffset`
`StageZOffset` *distance* Unit: *µm*

Sets a stage offset for all programmed x|y|z-coordinates. The offset is added to each x|y|z-coordinate of each programmed point. This command is only applicable in `StageScanMode`.

`GalvoXOffset`
`GalvoYOffset`
`GalvoZOffset` *distance* Unit: *µm*

Sets a galvo offset for all programmed x|y|z-coordinates. The offset is added to each x|y|z-coordinate of each programmed point. This command is only applicable in `GalvoScanMode`.

`MoveStageX`
`MoveStageY` *distance* Unit: *µm*

Triggers a relative stage displacement on the x|y-axis.

`GotoX`
`GotoY` *distance* Unit: *µm*

Positions the piezo or the stage to the defined x|y-coordinate, depending on the current `ScanMode`. For `GalvoScanMode` the positions will always be addressed with the piezo.

`StageGotoX`
`StageGotoY` *distance* Unit: *µm*

Positions the stage to the defined x|y-coordinate. This command ignores offsets and transformations.

`PiezoGotoX`
`PiezoGotoY`
`PiezoGotoZ` *distance* Unit: *µm*

Positions the piezo to the defined x|y|z-coordinate. This command ignores the offsets, but takes a possible transformation into account.

`CenterStage`

Moves the stage to the center coordinates (x=0, y=0) of the current sample position.

`StageVelocity` *velocity* Unit: *µm/s* Default: *2000*

Defines the displacement velocity of the stage given in µm/s, i.e., movements without exposure. Please note that the writing speed is defined separately via the `ScanSpeed` command.

`AddZDrivePosition` *distance* Unit: *µm*

Defines relative movements of the z-drive of the microscope. It is necessary to use this command for stitching structures higher than 300 µm but it is also suitable for layer by layer writing. Its direction is dependent on the `InvertZaxis` status.

B.4. Piezo PerfectShape parameters

Please note that all commands and settings related to PerfectShape are only relevant for the Piezo. The Galvo PerfectShape mode does not require any setting, but is fully automated and always on with NanoWrite 1.8.

`PerfectShape` *0, 1, 2, 3* Default: *2*

Sets the Piezo PerfectShape mode. Note that Galvo PerfectShape is always on.

- 0 PerfectShapeOff
- 1 PerfectShapeQuality
- 2 PerfectShapeIntermediate
- 3 PerfectShapeFast

`PerfectShapeOff`

Turns PerfectShape off for the Piezo. The conventional writing is recommended for straight lines and pointy corners. Note that the Galvo PerfectShape cannot be turned off.

`PerfectShapeQuality`
`PerfectShapeIntermediate`
`PerfectShapeFast`

Sets PerfectShape mode:
Quality: In this mode highest trajectory accuracy is achieved at rather low speed.
Intermediate: The trajectory accuracy is lower but the structures are written faster compared to `PerfectShapeQuality`.

Fast: Minimizes the writing time by maximizing the writing speed. The piezo's trajectory is less accurate than for `PerfectShapeIntermediate` and for `PerfectShapeQuality`.

`psLoadPowerProfiles` *path*

Loads a set of power profiles.

`psPowerProfile` *name*

Selects a stored power profile.

`psPowerSlope` *slope* Default: *0*

Sets the slope which is used for the laser power aberration compensation.

B.5. Loop statements and variables

`include` *file*

Adds the code of a separate GWL-file to the current GWL-file at the current command line. If the file resides within the current directory no path needs to be given. Otherwise the path can either be relative (`subdir\file.gwl`) or absolute (`D:\gwls\file.gwl`).

`repeat` #

Sets the number of repetitions for the next `include` line only. To repeat several lines of code, put them into the included file or use a loop. The total number of executions is $n + 1$, since the following line itself will also be executed.

`var` *$var_name = value*

Defines a new variable with an initial value. This value may be an integer, a decimal number or another variable, but it is not optional. In the latter case the value of the variable will be copied, there will not be a remaining connection between the variables. Variable names are case insensitive and may contain numbers and underscores.

`local` *$local_var_name = value*

Defines a new local variable with an initial value. The usage is exactly like var but the scope of the variable is limited to the current GWL file only.

`set` *$var_name = value*

Updates the value of an already defined variable. The same rules for valid values apply as with defining variables.

`if ... [elif ...] else... end`

Conditional jumps based on boolean expressions. If the condition is true the block following the `if` keyword is executed otherwise the block following the `else` keyword. `elif` is short for else if and can be used to chain conditionals.

Boolean expressions can be constructed using the comparison operators ==, !=, >, >=, <, <=, the unary operator Not and the binary operators And, Or. Boolean expressions can only be used in `while` loops and `if/elif` conditions. In particular their value can not be assigned to a variable. Please note that comparisons are subject to the fact that all variables are floating point numbers.

`for $i=a to b [step s] ... end` Default: $s = 1$

Loop command applied on a variable. The common syntax is as follows: for $i = a to b step s, where the variable $i starts at the value a and counts with increments of s to the value b. Step s is optional and has a default value of 1.0. ($a has to be a variable which is set within the for-line). The For-Loop must always be ended with end! You may also refer to chapter Variables, Loops & Calculation for more extensive examples.

```
local $i=0
local $start = 0
local $increment = 2
for $i=$start to $end step $increment
   %do something
end
```

`while boolean_expression ... end`

Like the for-loop but using a boolean expression for the loop condition. For an explanation of boolean expressions see if-else-elif. The loop keeps iterating as long as the boolean expression returns True. For avoiding infinite loops, the variables must be updated within the loop so that the boolean expression returns False at one point! Use while loops with care.

`continue - break`

Applicable in for and while loops. The break statement exits the loop immediately. The continue statement continues with the next iteration of the loop by jumping to the evaluation of the loop condition.

```
local $k = 0
while $k < 10
   set $k = $k + 1
   if $k == 2
      continue
   end
   if $k == 5
      break
   end
   TextPositionY 10*$k
   WriteText "New value %d" # $k
end
```

A local variable $k is used within a while loop. The boolean expression is tested at each iteration, until it returns False. In the loop, two conditional jumps are tested on the value of $k: if the first is true then the continue operates a jump to the next iteration, and if the second is true the break keyword exits the loop.

`Add`

The Add commands are adding the specified values to the current values of the respective command. Following Add commands are available:

`AddScanSpeed`	`AddLineNumber`
`AddLineDistance`	`AddPowerScaling`
`AddDefocus`	`AddLaserPower`
`AddExposureTime`	`AddUpdateRate`
`AddPointdistance`	`AddXOffset`
`AddYOffset`	`AddZOffset`

`Mult`

The `Mult` commands are multiplying the specified values with the current values of the respective command. Following `Mult` commands are available:

`MultScanspeed`	`MultPowerScaling`
`MultLineDistance`	`MultLineNumber`
`MultDefocus`	`MultLaserPower`
`MultExposureTime`	`MultUpdateRate`
`MultPointdistance`	`MultXOffset`
`MultYOffset`	`MultZOffset`

`#()`

The # character can be used for string formatting in order to convert the current value of a given value into a text string value. This value can be used in combination with any command requiring a string sequence as argument. The formatting syntax is the following, provided a variable $number was defined in the code previously: With the string sequence the % character sets the position of string formatted value within the string sequence, the conversion code gives its type and the character # indicates the variable to use. Syntax modifiers allow for personalized string formatting. The generally supported syntax for the target is the following:

%[−][+][ˆ][0][*width*][*.precision or _SignificantDigits*]ConversionCode where the ConversionCode can take the values d, u, f, e and g, for unsigned integer, signed integer, float, scientific notation and automatic float, respectively. The automatic float uses scientific notation only for exponents greater than 3. Please note that the square brackets shall not be written in the script. Example: `WriteText "radius = %_3f" #($number)`

B.6. Writing Text

`WriteText` *"text"*

This command can be used for descriptive text that will be exposed into the resist. That text has to be within double quotes. A linefeed can be introduced with \n.

Variables can used in this form: WriteText "float=%.2f, int=%d" #($f,$i)

For more string formatting details please refer to the user manual. Please note that `WriteText` has its own settings for writing speed, laser power and a fixed PowerScaling of 1.0.

`TextPositionX`
`TextPositionY`
`TextPositionZ` *distance* Unit: *µm* Default: *0*

Offset the position of the first letter of the last line in x|y|z-direction.

LineSpacingX
LineSpacingY
LineSpacingZ *distance* — Unit: *µm* — Default: *0*

Offset in x|y|z-direction that will be added each time a linefeed is invoked (\n). Please note that with negative y-values for LineSpacingY the direction of the linefeed can be changed.

TextLaserPower *0-100* — Unit: % — Default: *30*

Laser power value that is used for writing text. It is independent of the setting of the normal LaserPower and does not influence that parameter. Please note that for TextLaserPower the value of PowerScaling is always internally set to 1.0.

TextPointDistance *distance* — Unit: *nm* — Default: *50*

Sets the distance between two interpolated points for writing text. Please note that the unit is nanometer!

TextScanSpeed *speed* — Unit: *µm/s*

Sets the speed for writing text.

TextFontSize *fontsize* — Unit: *µm* — Default: *5.0*

Sets the size of the font used for the WriteText command. The size is given in µm and corresponds to the distance between baseline and cap height.

B.7. Correction features

DefocusFactor *factor* — Default: *1*

Each z-coordinate is multiplied with this factor. The DefocusFactor compensates for shifts of the focus position in z due to the index mismatch between the immersion medium, the substrate and the photo resist.

MeasureTilt #

Specifies the array size of the measurement of the sample tilt. MeasureTilt 5 measures the interface on a regular 5x5 array on the substrate, thus at 25 points throughout the maximum writing area. Hence, in PiezoScanMode the measurement area is 300 µm^2, in StageScanMode the measurement area is the maximum writing area on the current sample which is specified in the Exchanger.ini.

TiltCorrectionOn
TiltCorrectionOff

TiltCorrectionOn activates tilt correction based on previously measured values by MeasureTilt. The tilt correction compensates for substrate tilts by adapting the z-position for each x-y-coordinate individually. The angle of the z-axis will not be changed.

Before this command there should be a MeasureTilt command to have valid values. Subsequently a FindInterfaceAt command should always be performed. Please note that the GalvoScanMode does not feature a tilt correction.

TiltCorrectionOff deactivates tilt correction.

`ManualTiltX`
`ManualTiltY` *angle* Unit: °

`ManualTiltX` sets values for tilt correction for x-axis. `ManualTiltY` sets values for tilt correction for y-axis.

`AccelerationTime`
`DecelerationTime` *time* Unit: *s*

Sets the time interval in which the laser power is adjusted at a line-start (line-ending for `DecelerationTime`), to account for the acceleration of the piezo. Neither valid for `PerfectShape` nor `GalvoScanMode`.

`Acceleration`
`Deceleration` *exponent*

Defines the shape of the laser power adjustment curve at a line-start or line-end. Neither valid for `PerfectShape` nor `GalvoScanMode`.

B.8. Autofocus parameters

`FindInterfaceAt` *z-value* Unit: *µm*

Triggers an autofocus procedure with the given value as z-coordinate of the piezo. This value is a hardware offset of the interface. That way coordinates below the interface may be accessed and the structures can be well anchored to the substrate surface.

After a successful search, the Message Log displays: Interface found x@y. The first interface value x is a measure for the interface signal amplitude, the second interface value y is inverse proportional to the exposure time of the autofocus camera and is a value in between 1 and 100. Thus $y = 1$ means maximum exposure time, $y = 100$ means minimal exposure time. These two interface values are useful for choosing the proper interface if multiple interfaces are present.

The signal strengths and amplitudes depend very much on the index mismatch between the two adjacent media, the objective used and the actual settings of the autofocus camera. At an air-glass interface the signal amplitude may be larger than 1000 and the exposure time will be very short (>90), whereas at a glass-resist interface the signal amplitude x is in general between 1 and 400 (depending on the resist), and the exposure time is often at maximum (y=1).

`InterfaceMax` *x.yyy*

Sets a maximum value for the interface. Consider the Message Log displaying *Interface found x@y*. `InterfaceMax x.yyy` suppresses all interfaces with larger amplitude or with shorter exposure time (higher second interface value).

Example: `InterfaceMax 100.002` suppresses all interfaces with an amplitude > 100 at an exposure time value > 2. Please note that the exposure time value has to be specified with three digits! InterfaceMax and InterfaceMin cannot be set by the same time. The last entry will be used.

`InterfaceMin` *x.yyy*

Sets a minimum value for the interface. Consider the Message Log displaying *Interface* *found x@y*. `InterfaceMin x.yyy` suppresses all interfaces with lower amplitude or with longer exposure time (lower second interface value).

Example: `InterfaceMin 100.002` suppresses all interfaces with an amplitude < 100 at an exposure time value < 2. Please note that the exposure time value has to be specified with three digits! InterfaceMax and InterfaceMin cannot be set by the same time. The last entry will be used.

`ResetInterface`

Erases `InterfaceMax` or `InterfaceMin` settings.

`InterfacePosition` *z-value* Unit: *µm*

NanoWrite applies the `DefocusFactor` starting from the z-value given in the most recent `FindInterfaceAt` command. `InterfacePosition` overwrites this value so that the `DefocusFactor` can be used starting at any desired z-coordinate. This command does not influence the interface finding routine.

B.9. Initialization parameters

`SamplePosition` #

Moves to the defined sample position: The objective moves down, the stage addresses the defined sample center and an approach procedure is performed. This is different from double clicking on a sample position.

`ChooseObjective` #

After lowering the objective, the specified objective number is addressed by the motorized objective turret of the microscope. The objective name is displayed in the Message Log. There will not be an automated approach to allow for a prior sample position change.

`InvertZAxis` *0, 1* Default: *0*

Triggers the inversion of the z-axis. To conserve a right-handed coordinate system the x-axis is inverted at the same time.

0 deactivates InvertZAxis
1 activates InvertZAxis

B.10. Protocol and logging

`CapturePhoto` *"filename.tif"*

This command will save a TIFF file with the current live view image. The camera view in NanoWrite has to be active for this feature to work. Captured images may only be visualized with certain programs that support the used 16 bit image format which is not the case for Windows 7 standard viewers. However, most freeware software does support this feature. It may be useful to set a preceding `Wait 0.5`, if stage movements are performed before. Otherwise the images may be blurry. Furthermore, activating auto-exposure and auto-contrast are recommended.

`TimeStampOn`
`TimeStampOff`

For `TimeStampOn` each message in the Message Log is marked with the current time. For `TimeStampOff` no current time stamp is added. These two commands can be used anywhere in the code.

`MessageOut` *"text"*

Displays the text within quotation marks in the Message Log.

`DebugModeOn`
`DebugModeOff`

Messages due to `MessageOut` and `ShowVar` will be displayed in the Message Log when loading jobs.

`ShowParameter`

Displays the power adaption parameters as well as the current `InterfaceMin/Max` value in the Message Log. Please note that `PerfectShape` and `GalvoScanMode` use different approaches.

`ShowVar` *$variable*

Displays the name and value of the given variable in the Message Log.

`SaveMessages` *"path"*

Saves the current content of the Message Log in the file specified.

`Pause`

Triggers a pause in the writing process and opens a dialog window that has to be confirmed by the user to proceed with the writing process.

`ZDrivePosition`

Writes the current position of the microscope z-drive to the Message Log.

`NewStructure`

Pressing the skip-button on the NanoWrite GUI (advanced tab) during the writing procedure makes the system stop the current writing process and resume writing after the the next `NewStructure` command. Please note that `MoveStage`, `StageGoto` and `FindInterfaceAt` commands will be executed anyway.

B.11. Maintenance

`ManualControl`

Opens the manual control window. In this window the controls for the AOM voltage and for the electric shutter are directly accessible. Furthermore the photo diode voltage is plotted. Do not use this window unless you are sure what you are doing!

`ReloadIni`

Reloads the INI-files of the system. That way a change in the INI-files can be directly taken into account without closing and restarting the NanoWrite software. Please note that for some values, especially concerning the exchanger.ini a NanoWrite restart is still necessary.

`Recalibrate`

Recalibrates the laser power, i.e., measures the power scale over AOM voltage.

B.12. Standard header

Standard header PiezoScanMode

```
PiezoScanMode
ContinuousMode
ConnectPointsOn
PerfectShapeIntermediate
%PerfectShapeOff  % writing without PerfectShape
%ScanSpeed 100    % writing without PerfectShape

%InvertZAxis 0  % writing in standard configuration
%InvertZAxis 1  % writing in Dill configuration
%MeasureTilt 5
TiltCorrectionOff %TiltCorrectionOn %choose as desired

PowerScaling 1.0
LaserPower 30

XOffset 5
YOffset 5
ZOffset 0
```

Standard header GalvoScanMode

```
GalvoScanMode
ContinuousMode
ConnectPointsOn

ScanSpeed 10000

%InvertZAxis 0  % writing in standard configuration
%InvertZAxis 1 % writing in Dill configuration
TiltCorrectionOff % not available with GT

PowerScaling 1.0
LaserPower 50

XOffset 0
YOffset 0
ZOffset 0
```

Index