VxCraft 1.0

Usermanual

TABLE OF CONTENTS

PROLOGUE

Welcome to the VxCraft manual! This software is a powerful tool for creating NC programs for CNC machines. With a few mouse clicks you can convert your graphics into complex 3D models or load your existing 3D models. VxCraft offers numerous possibilities to convert these models into executable NC programs and to optimize your CNC machining process.

This manual provides a comprehensive introduction to the many features and tools of VxCraft. It explains the various aspects of the software, from basic operation to loading and modifying models, creating tools, calculating and simulating toolpaths, and exporting using customizable or custom postprocessors. The goal is to give you a solid understanding of the software and help you use VxCraft effectively.

We hope that this manual will help you to use the software efficiently and wish you much success in your projects with VxCraft!

Requirement

The effective use of VxCraft and understanding of this manual requires a basic knowledge of CNC machining and CAM software. Although this manual is designed to guide you through the specific features and processes of VxCraft, it is not designed to be a comprehensive course in CNC/CAM. If you are not already familiar with the basics of CNC machining, we recommend that you first acquire a basic knowledge of the subject.

.

Editions

VxCraft Free Edition
The Free Edition provides basic functions that are especially suitable for beginners or users with simple projects. You can create 3D heightmaps from graphics and use the roughing, contour and engrave job. However, you are restricted to a limited number of models, tools, and jobs per project. Features such as multi-sided machining and advanced job parameters are not available. In addition, powerful features such as radius correction and collision checking are already included in this edition.

VxCraft Pro Edition
With the Pro Edition, all objects can be freely placed and rotated, so that 3-axis editing of the scene is possible from all directions. Furthermore, the Pro Edition contains all the functions of the Free Edition and extends it with numerous additional features.

Product features

Objects	Free	Pro
3D highmap from graphic (*.bmp, *.png, *.jpg)	✓	✓
Freely definable stocks	✓	✓
Maximum object size adjustable	no	✓
3D models (*.stl)	no	✓
Converter: 3D model to 3D heightmap	no	✓
Converter: 3D heightmap to 3D model	no	✓
Converter: Heightmap impression from 3D model	no	✓
Number of models per project	1	unlimited
Number of tools per project	2	unlimited
Number of stocks per project	1	unlimited
Number of jobs per project	2	unlimited

Positioning and multi-sided machining	Free	Pro
Multi-sided machining	no	✓
Position NC-Zero Point(all axes)	✓	✓
Rotate NC-Zero Point (Z-axis)	no	✓
Rotate NC-Zero Point (all axes)	no	✓
Scale models	✓	✓
Position models	✓	✓
Rotate models (Z axis)	no	✓
Models rotate (all axes)	no	✓

Job strategies	Free	Pro
Job strategie "Roughing"	✓	✓
Job strategie "Contour"	✓	✓
Job strategie "Engrave"	✓	✓
Job strategie "3xShape"	no	✓
Job strategie "Drilling"	✓	✓

Job parameters	Free	Pro
3D Radius correction	✓	✓
Collision check	✓	✓
Roughing microstep	no	✓
Bounding Model Reference	no	✓
Zone limitation	no	✓
Spiral plunge	no	✓
Ramp plunge	no	✓
Geometry allowance	no	✓

Tools	Free	Pro
Flat End Mill	✓	✓
Ball End Mill	✓	✓
Graver	✓	✓
Graver Flat End	no	✓
Graver Round End	no	✓
Bull End Mill	no	✓
Roundover	no	✓
Shank geometry for tools	no	✓
Freely definable tools (Lua-Script)	no	✓

Program modules	Free	Pro
freely programmable postprocessors (Lua-Script)	✓	✓
Solid Live Simulation	no	✓
Create and manage templates	no	✓

Miscellaneous	Free	Pro
Commercial use	no	✓

Labels in the manual

In this manual, certain functions and, where appropriate, entire chapters are labeled with special markers that either indicate usable parameters or denote the availability of a function for a particular edition.

Postprocessor parameters

In some positions in the manual, the parameter names for the postprocessor of the respective topic can be found in small print, in square brackets.

> ***Example:***
>
> - **Spindle speed** *[ppSpindlespeed]*

This makes programming a postprocessor easier, as the desired parameter can be found very easily via the help functions.

Note to the Edition

Functions or chapters that are only available for VxCraft Pro Edition are marked with <PRO>.

.

> ***Example:***
>
> **Solid Live Simulation<PRO>**

System requirements

For optimal performance and stability of VxCraft, you should ensure that your computer meets the following minimum requirements. The recommended requirements ensure that you can use VxCraft to its full potential, especially when working with complex models and large projects. Hardware below the minimum requirements could lead to performance degradation and possible instability.

Requirement	Minimum requirements	Recommended
OS	Windows 7	Windows 10
Processor	64-bit Processor with 2,2 GHz	64-bit Processor with 3,4 GHz
Memory	1,5 GB	3 GB
Graphic	Intel(R) HD Graphics 530 Graphics memory: 1GB Support for OpenGL 3.3 or higher	NVIDIA GeForce GTX960 Graphics memory: 2GB
Installation size	50Mb	100Mb
Internet connection	optional for updates and additional downloads, as well as for viewing the online manual	
Default browser (Manual)	To view the local manual, a common, modern browser with Javascript enabled should be set as the operating system's default browser.	
Screen resolution	1280 x 720	1920 x 1080 (Full HD)
Input devices	Mouse, Keyboard	

Installation

The installation of VxCraft is a simple and fast process guided by a setup wizard. Before starting the installation, there is one important requirement to keep in mind.

Copy protection dongle<PRO>

In the VxCraft Pro Edition, a licensed copy protection dongle is used as part of the security measures. Please make sure that the supplied dongle is correctly connected to your computer. The dongle will be automatically detected and installed as soon as it is connected.

Current version

Before installing, it is recommended to check the website www.vxcraft.com to see if a more recent version of VxCraft is available.

Installation process

Start the downloaded or included setup wizard and follow the instructions. At the beginning of the installation process, you have the option to select the desired language. This selection determines the language of the program interface. Please note that you can change this setting at any time after the installation is complete in the **Program Settings**. After the installation is complete, you can start and use VxCraft immediately.

Programdata folder

Depending on the installation type (administrator mode or single user mode) of VxCraft, a specific folder is created. This folder contains important files such as templates, settings, sample projects and crash logs. Please note that VxCraft requires write permissions for this folder.

Installation type	Folder path	Example
Administrator mode	%programdata%\VxCraft	C:\ProgramData\VxCraft
Single user mode (does not require admin rights)	%AppData%\VxCraft	C:\Users\<Username>\AppData\Roaming\VxCraft

Updates

Regular updates for VxCraft are available on our website www.vxcraft.com. These updates often contain important enhancements, including security-related updates, and we therefore recommend that you always keep VxCraft up to date.

Before installing an update, we recommend backing up the **ProgramData folder**. This folder contains important files such as templates, settings, and sample projects. By backing up this folder, you ensure that you can fall back on your previous data in case of complications during the update process.

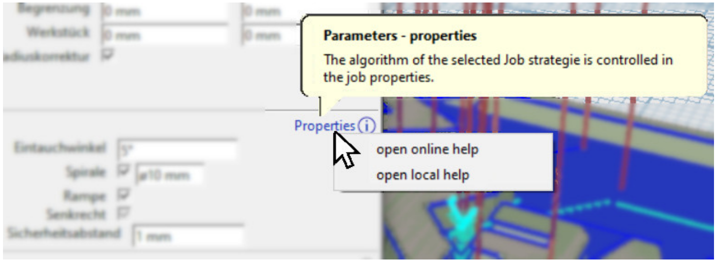
USER INTERFACE (GUI)

The user interface, also known as the Graphical User Interface (GUI), is the central aspect of any program to allow efficient and intuitive interaction between the user and the software. This chapter covers the various elements of the GUI, starting with general elements relevant to all areas, followed by the main window and smaller, more specific dialog boxes. Finally, the available key combinations (shortcuts) for faster and more efficient operation of the user interface are presented.

General GUI elements

Quickinfos

Quickinfos are small, blue info icons (a small "i" in a circle) distributed throughout the program to provide you with additional information about the respective elements. When you move the mouse pointer over such an info icon, a short description about the topic is displayed as a tooltip. Clicking on the info icon will take you to the detailed explanation in the manual. You have the option to choose between the online manual and the local manual, depending on your preferences and the availability of an Internet connection.



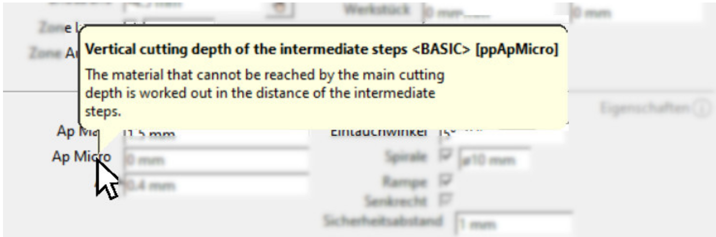
Parameters - properties

The algorithm of the selected Job strategie is controlled in the job properties.

© vxcraft.com

Tooltips

Tooltips are seamlessly integrated into the program and appear when the mouse pointer hovers over icons, labels, titles and other common GUI elements. They provide a brief explanation of the function or meaning of the element in question. If you need more information, look for the nearest tooltip icon.



Vertical cutting depth of the intermediate steps <BASIC> [ppApMicro]

The material that cannot be reached by the main cutting depth is worked out in the distance of the intermediate steps.

Numeric input fields

In numerical input fields, both direct numerical values and simple mathematical formulas can be entered. To change the value of a field, hold down the left mouse button and move the mouse up or down. This function is not available for formulas and can be disabled in the settings if desired. Alternatively, you can use the arrow and scroll keys to increment the value. To confirm the entered value or formula, press Enter or F5. When the focus leaves the input field, the input is also confirmed.

Input fields

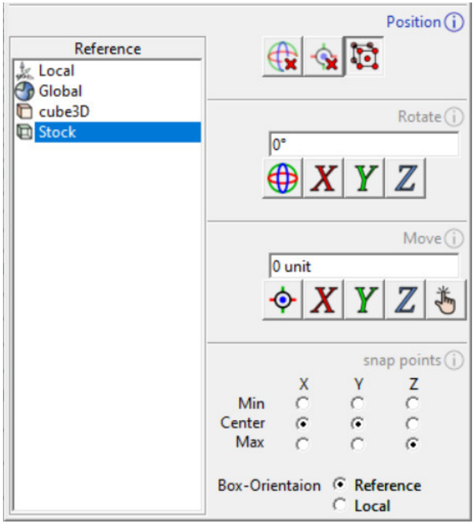
In the Script window, you can enter text to run scripts. To validate the entered text and execute the script, press Enter or F5. When the focus leaves the input field, the script will also be validated and executed.

GUI-Element "Status bar"

The status bar is a GUI element that displays general information about the mouse coordinates and other info. Optionally, there is a progress bar on the right side, which offers the possibility to cancel the current calculation. During a calculation the progress bar is activated and the title of the current calculation process appears to inform the user about the progress of the calculation.

Tab "Positioning"

Position and align objects (Models, stock, NcPack/NC-Zero Point,...) in 3D space using positioning tools and references.
To do this, various tools and settings are available that influence or reset the behavior of the positioning.



Reference
- Local
- Global
- cube3D
- Stock

Rotate: 0°

Move: 0 unit

snap points

	X	Y	Z
Min	○	○	○
Center	●	●	○
Max	○	○	●

Box-Orientaion ● Reference ○ Local

Tab "Positioning"

Reference list

Select a reference to align the object to, or use special references for a local or global axis movement.

The reference list makes it possible to select a reference as a positioning aid. By selecting a reference, the object can be aligned to the reference or its axes can be used for the direction of movement. In addition, there are two special references:

- **Local:** When this special reference is selected, the object is moved and rotated along its own axes.
- **Global:** When this special reference is selected, the object is moved along the global axes and rotated.

However, the object to be positioned is not bound to the selected reference, so its position and rotation remain unchanged even if the reference changes or disappears.

Activate snap points

Displays a table of snap points of the reference that can be used for precise positioning of the object.

For more information on snap points and how to use them, see the **"Snap points"** section in the manual.

Reset position
Resets the position of the object to the global origin.

The object is moved to the starting point of the 3D space and all previous positioning is discarded. The object rotation remains.

Reset rotation
Resets the rotation of the object to the global axis orientation.

All previous rotations are discarded and the object is displayed in its original orientation according to the global coordinate axes.

Rotate

<PRO>

Rotates the object by the specified number of degrees relative to the selected axis of the reference. Right-click on the desired axis to rotate the object in the opposite direction.

The "Rotate" function allows to rotate the object around the actuated axis (X, Y<PRO> or Z <PRO>) in the number of degrees entered in the input field. The rotation is relative around the selected axis of the reference. If the "local" tool is selected in the reference list, the object uses its own axis orientation, if "global" is selected, it uses the global axis orientation. If an object is selected as a reference, the axis orientation of the reference object is used.

Align rotation
Aligns the rotation of the object with the reference.

The "Align Rotation" function transfers the rotation of the reference to the object so that the object points in the same direction as the reference. The position of the object is not changed. This function is useful if the object is to be adapted to certain orientations of reference objects.

Move

Moves the object relatively by the specified value along the selected axis of the reference. By right-clicking on the desired axis, the object can be moved in the opposite direction.

The "Move" function enables the object to be moved along the actuated axis (X, Y or Z) by the value entered in the input field. The displacement is relative along the selected axis of the reference. If the "local" tool is selected in the reference list, the object uses its own axis alignment, if "global" is selected, it uses the global axis alignment. If an object is selected as a reference, the axis orientation of the reference object is used.

Align position

Aligns the position of the object with the selected reference.

The "Align Position" function allows you to align the position of the object with the selected reference. With local reference this function has no effect. If "global" is selected in the reference list, the object is set to the origin of the 3D space (world origin). If an object is selected as reference, the position of the object is set to the reference. Both objects share the same position in this case. Positioning does not affect the rotational orientation of the object.

Place object

Places the object at the selected target point in the render window. An alternative origin point can be selected by holding the Ctrl key.

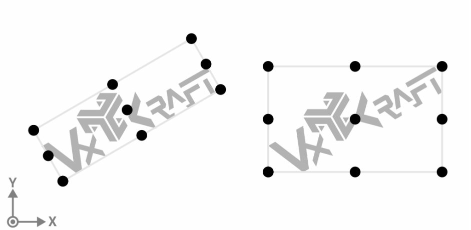
The "Place Object" function allows you to place the object at a target point selected in the render window. In the default mode, the object origin jump is set directly to the clicked target point. In alternative mode, which is activated by holding down the Ctrl key, an alternative object origin jump can be selected with the right mouse button before the target point is set with another click. The Esc key can be used to cancel the selection at any time.

Snap points

By means of the snap table, the object origin can be set at a snap point of an imaginary box enclosing the reference object.

Three values are available for each of the three axes (X, Y and Z): the minimum (Min) of the box, the center (middle) of the box and the maximum (Max) of the box.

Box orientation:
Reference: The box is oriented to the selected reference.
Local: The box is oriented to the object to be positioned.



left: *Box orientation: Reference*
right: *Box orientation: Local*

Tab "Rendering"

Controls the appearance of the object in the render window. Not every object has access to all of the options listed here.

Line style

Controls the line style of the object.

Style	Description
Solid	The object is rendered as a solid line.
Dots	A dot is rendered at the beginning and end of each line segment. The line itself is hidden.
Solid + Dots	The two modes line and points are rendered together.

Color

Colors the model geometry with the set color.

Cutting edge

Colors the cutting edge with the set color.

Color rapid

All rapid movements in the calculated toolpath are displayed with this color.

Color feed rate

All segments in the calculated toolpath with milling feed rate are displayed with this color.

Accent color

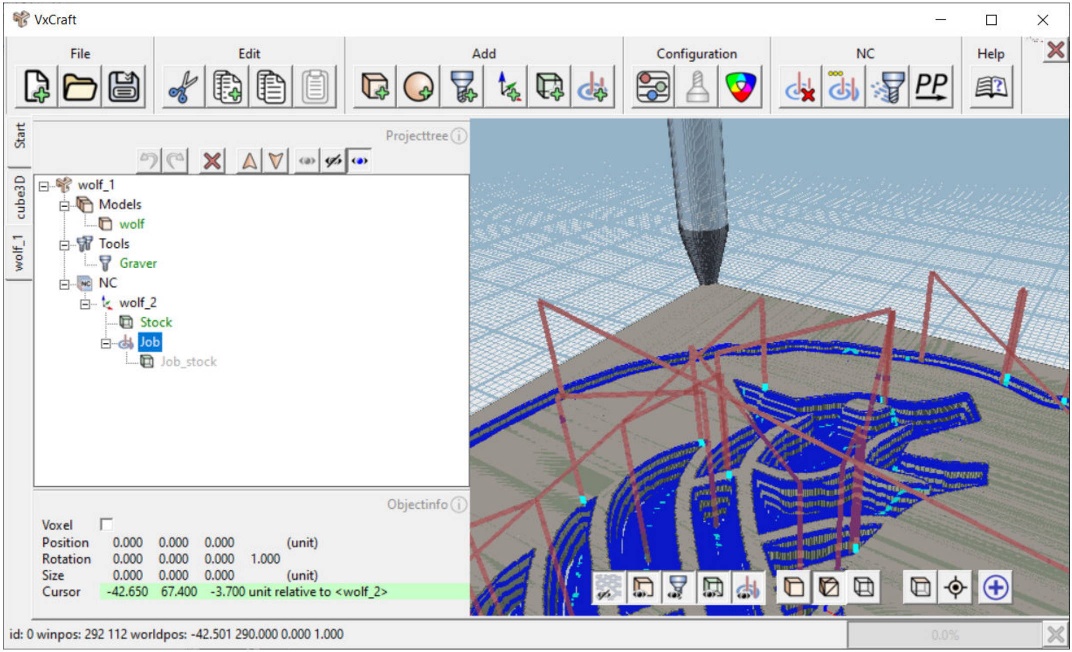
Highlights special features of the object.

Model color simulation

Determines the color of the resulting stock, as well as the simulation ablation model.

Main window

The main window of VxCraft is the central workspace of the application, where all important functions and tools are provided. It includes the rendering window for displaying 3D objects and scenes, the menu bar for accessing main functions, project tabs of the opened projects, the project tree for organizing objects and layers, and the object info box that displays information about selected objects.



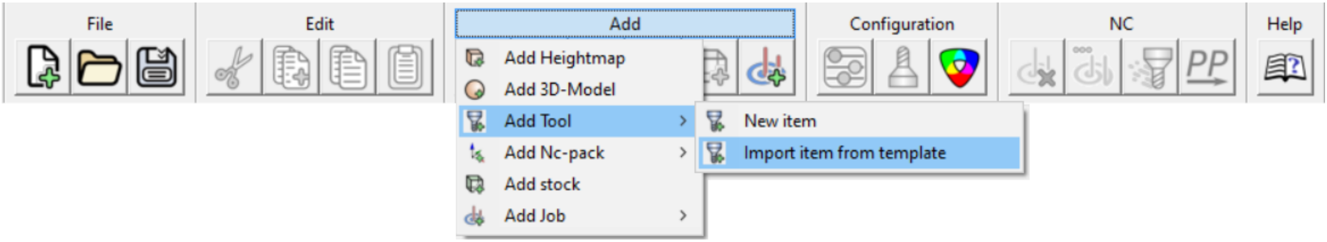
Voxel □
Position 0.000 0.000 0.000 (unit)
Rotation 0.000 0.000 0.000 1.000
Size 0.000 0.000 0.000 (unit)
Cursor -42.650 67.400 -3.700 unit relative to <wolf_2>

id: 0 winpos: 292 112 worldpos: -42.501 290.000 0.000 1.000

Main window

Main menu

The main window has a menu at the top that provides access to the various functions and settings of the program. It is divided into logically structured sections. Each category in the menu has a title and the most important functions as icons for quick access. Clicking on the category name opens a popup menu containing all the functions of that category.



Main menu

Menu category "File"

This menu group contains the file management for input and output, as well as the general control of the program.

New
Creates a new project and starts with an empty working environment.

Open
Opens an existing project or template.

Save
Saves the current state of the open project or template.

Save as
Saves the currently opened project or template under a new file name.

Close project
Closes the currently opened project.

Close
Asks whether the opened projects should be saved and exits VxCraft.

Menu category "Edit"

This menu group contains general functions for selected objects.

Cut
Removes the selected objects from the project tree and copies them to the clipboard.

Duplicate
Creates a copy of the selected objects.

Copy
Copies the selected objects to the clipboard. The child objects, unless also selected, are not copied.

Paste
Pastes the objects from the clipboard into the project.

Remove
Removes the selected objects from the project.

Remove all references
Deletes all connections of the selected objects to other objects. Useful when inserting objects from another project.

Menu category "Add"

This menu group contains functions for creating and importing new objects.

Add heightmap
Adds an empty heightmap object. To create the geometry, various options are available in the configuration dialog, which opens automatically after creation.

Add 3D model
Adds an empty 3D model object. To create the geometry, various options are available in the configuration dialog, which opens automatically after creation.

Add tool
Adds an empty tool object or loads one from a template. The geometry is defined using the tool editor, which opens automatically after an empty tool is created.

Add NC-Pack
Adds an empty NC-Pack object or loads one from a template. An NC-Pack defines the zero point for the jobs it contains. The zero point can be set in the configuration dialog, which opens automatically after an empty object is created.

Add stock
Adds an empty stock object. To create the geometry, various options are available in the configuration dialog, which opens automatically after creation.

Add job
Adds an empty job object or loads one from a template. Contains all parameters for calculating a toolpath. The parameters can be defined in the configuration dialog, which opens automatically after creation.

Menu category "Configuration"

This menu group contains functions for configuring selected objects and the color palette.

Configuration object
Opens the corresponding configuration dialog for the selected object, depending on the object type.

Tool editor
Opens the tool editor for the selected tool to edit or define the geometry.

Colors
Opens the dialog with the color palette for creating and changing colors.

License manager
Opens a dialog window for managing the purchased license. The license manager displays the current license status and can be used to load or remove a purchased license.

System settings
Opens a dialog window for adjusting the system settings. Once this window is closed, all changes made are automatically saved. Some settings require a restart of the program for the changes to take effect. For more information see the chapter **"Lua module settings/settings.lua"**.

Menu category "NC"

The NC menu group contains functions for managing the toolpaths

Reset toolpath
Deletes the calculated toolpath. The job object itself is retained.

Calculate tool path
Deletes the existing toolpath if necessary and calculates the job object. Creates a toolpath if successful.

Solid Live Simulation
Starts the environment "VxCraft - Solid Live Simulation". This can be used to simulate the selected toolpaths.

Postprocessor
Starts the postprocessor environment

Menu category "Help"

The Help menu group contains functions for displaying documentation and information about VxCraft.

Open user manual
Opens the table of contents of the user manual. You can choose between the online version and the local version.

Program files
Opens the folder in Explorer where VxCraft stores temporary data, settings files, templates, sample projects and postprocessors.

Update
Opens the VxCraft information dialog, which contains a link for checking for available updates.

About VxCraft
Displays basic information about VxCraft.

Project tab

In the main window, vertically arranged tabs can be found on the left side for each open project. By clicking on one of these tabs, the active project can be changed.

Project tree

The project tree is a central element in VxCraft's workflow and displays the structure of the entire project. The context menu opens by right-clicking on one or more selected objects, dynamically adapts to the selection and offers almost all the functions of the main menu.

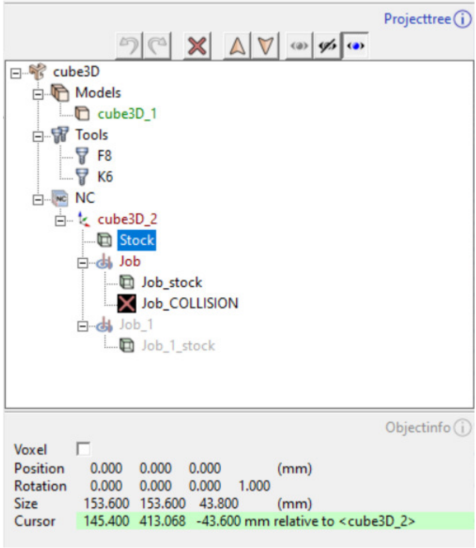
Object types and structure

This section gives an overview of the object types and the tree structure of a project.

The project tree is hierarchical and has a fixed structure. The topmost object is the project element and bears the name of the saved project file. The project element contains all other objects.

Fixed folders

Each project has fixed folders from the beginning:



Voxel □
Position 0.000 0.000 0.000 (mm)
Rotation 0.000 0.000 0.000 1.000
Size 153.600 153.600 43.800 (mm)
Cursor 145.400 413.068 -43.600 mm relative to <cube3D_2>

Project tree

Folder	Description
Models	Includes all heightmaps and models of the project
Tools	Includes all tools of the project
NC	Contains all NC-Packs of the project

Creatable objects

The following table lists all the objects that can be created:

Object	Parent element	Short description
Heightmap	Models	A 3D geometry generated from 2D data
3D-Models	Models	A 3D geometry consisting of facets
Tool	Tools	Contains the cutting and shank geometry of an NC tool, as well as tool-relevant parameters.
NC-Pack	NC	Determines the NC-Zero Point for the containing jobs
Stock	NC-Pack	A heightmap that specifies the geometry of the starting material (blank) for the calculation.
Job	NC-Pack	A job object contains all references and parameters for the toolpath calculation.

Generated objects

The following table lists all automatically generated objects:

Object	Parent element	Short description
Resulting stock	Job	Is calculated and generated from the tool and stock reference of the job and the toolpath. Can be used as a stock reference for subsequent jobs.
Collision evaluation	Job	This optional object is created when a collision is detected during the toolpath calculation. It displays the collision points.

Text color

The text color of the elements in the project tree gives conclusions about the state and properties.

Text color	Description
blue	The object is referenced by a selected object.
red	The object references a selected object.
Faded color	The object is hidden.

References

References are links between objects in the project tree. An object can read the data and properties of the reference, but cannot change them. To prevent reference loops, some reference types only allow references that are above the object in the hierarchy.

Object name
The name of the object may occur only once in the project. If the name is already assigned, a consecutive number is appended to the name.

Project tree functions

Undo
Undoes the last change to the hierarchy.

Redo
Restores the previous change to the hierarchy.

Remove
Removes the selected objects from the project.

Up
Moves the selected object up in the hierarchy.

Down
Moves the selected object down in the hierarchy.

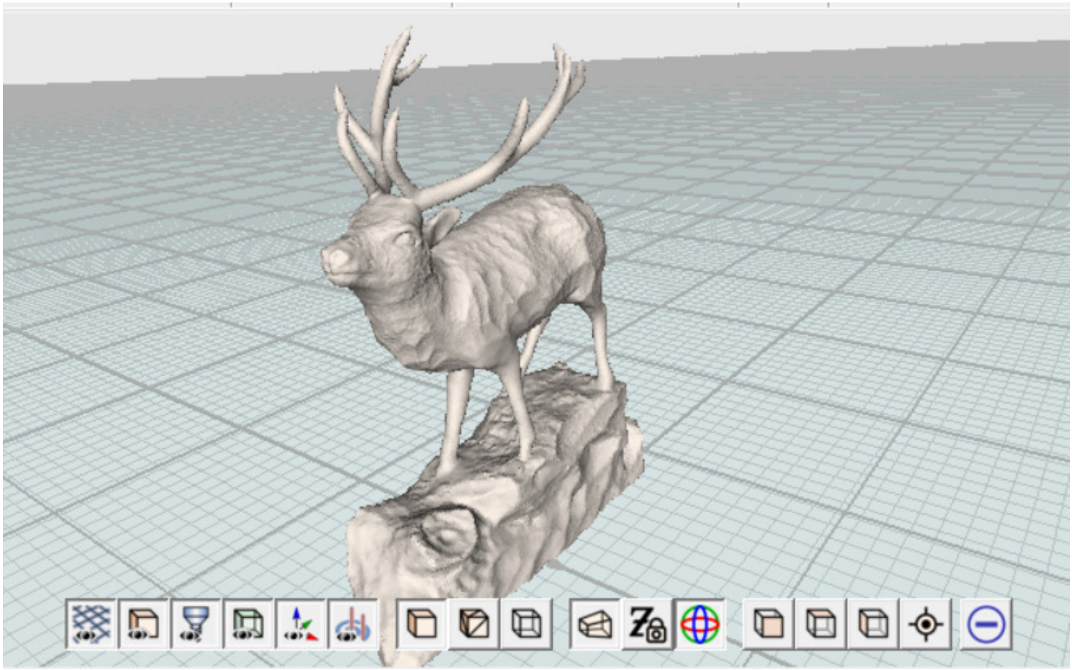
Show
Displays the selected objects. In order for the object to actually be shown in the render window, it must also be ensured that the corresponding layer is shown in the render window toolbar. The <> key can be used to toggle the visibility of an object.

Hide
Hides the selected objects. An object remains hidden even if the corresponding layer of the object type is shown in the Render Window toolbar. The <> key toggles the visibility of an object.

Automatically show and hide jobs
When this mode is activated, selected jobs are automatically faded in. Jobs that are not selected are automatically hidden.

Render window

The render window is the central visualization element in VxCraft. It shows the 3D rendering of the objects contained in the project tree and allows interaction and navigation in the scene.



© vxcraft.com

Render window

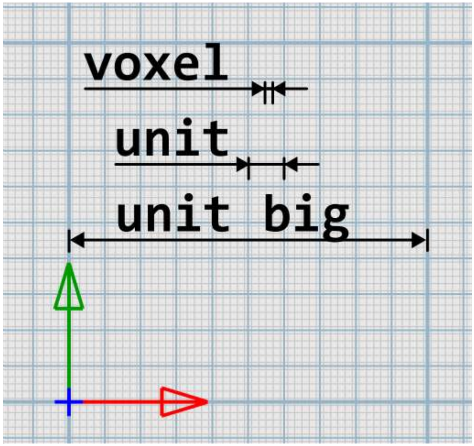
Workplane

Absolute zero point

The absolute zero point is represented as a coordinate system and is the origin of the project scene. It determines the three main axes X, Y and Z. The camera control and the working grid are oriented to this zero point.

Work grid

The work grid in the render view is located on the Z-plane of the absolute zero point. The size of the work grid can be used as a guideline for the maximum size of the models. It dynamically adapts to the project resolution and is based on the following scales:



© vxcraft.com

Workplane

Grid	Rendering	Scale	Comment
Voxel	fine lines	1 Voxel	Offset by half a voxel to absolute zero.
Unit	standard lines	1 User unit	Represents the distance 1 voxel x project resolution
Unit big	bold lines	10 User units	Represents the 10-fold distance of a unit. The factor can be changed in the settings.

Each of the individual grids is automatically faded in and out depending on the camera zoom level. In the chapter **"Project settings"**, the units of VxCraft are described in more detail.

Toolbar

The toolbar is located at the bottom of the render window and provides functions to control rendering and camera control. On the far right of the toolbar is a blue plus sign in a circle. When the plus is activated, the toolbar expands to include several additional functions. All functions are listed below, with the extra functions marked with a (+).

Layer group

The "Layer" symbol group controls the visibility of individual object groups. Regardless of the layer status, objects always remain hidden if they are hidden in the project tree (see also **Project tree functions**).

Layer	Description
Workplane	Hides/shows the grid of the workplane and the coordinate system of the absolute zero point
Model	Shows/hides heightmaps and 3D models
Tool	Hides/shows tools
Stock	Fades stock in/out
(+) NC-Pack	Shows/hides the coordinate systems of the NC-Zero Points
Job	Hides/shows jobs

Group rendering

Controls the general rendering mode of the objects.

Rendering	Description
Solid	Shows the objects as solid surfaces
Solid+Wire model	Shows the objects as solid surfaces with visible wireframe over them
Wire model	Shows the objects in wireframe mode

Group camera

Controls the behavior of the camera and the projection mode. The key assignment for the camera control is described in the chapter **Shortcuts**.

Function	Description
(+) Projection mode	Switches between perspective and orthogonal projection
(+) Z-Lock	Disables moving the camera target point in direction Z. "3D Rotation" is automatically disabled.
(+) 3D-Rotation	The camera allows rotation around two axes. By activating the 3D rotation mode, the camera control is extended so that rotation around all three axes is possible. "Z-Lock" is automatically deactivated.

Group view

Sets the camera to a specific orientation.

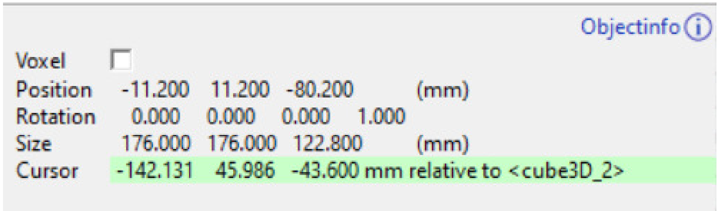
View	Description
(+) Front	Shows the scene from the front. Right click to show the scene from behind
Top	Shows the scene from above. Right click to show the scene from below
(+) Right	Shows the right side of the scene. Right-click to show the left side of the scene
Reset	Resets the camera to the original orientation

Additional functions

Shows and hides additional functions in the toolbar.

Object info

The object info area shows information about selected objects, such as position, rotation, size and cursor coordinates.



```
                                                Objectinfo ⓘ
Voxel      □
Position   -11.200    11.200   -80.200          (mm)
Rotation     0.000     0.000     0.000   1.000
Size       176.000   176.000   122.800          (mm)
Cursor    -142.131    45.986   -43.600 mm relative to <cube3D_2>
```

Object info box

Unit Voxel
With the help of this checkbox, the values displayed in the object info box can be changed to the unit "voxel".

Position
The position information gives the coordinates of the selected object with respect to the global origin.

Rotation
The rotation information shows the rotation values of the selected object as a quaternion with respect to the global origin.

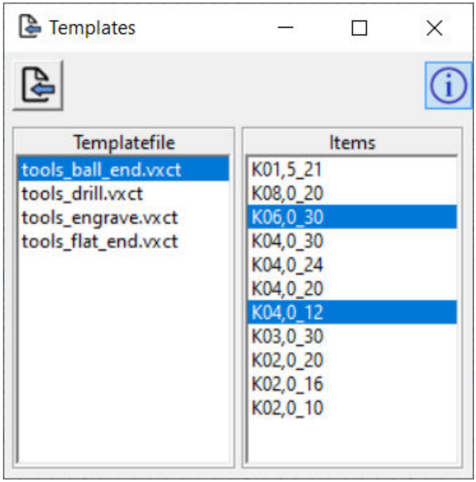
Size
The size values reflect the local size of the selected object.

Cursor
The cursor coordinates indicate the 3D position of the mouse in the render window. The origin for the position determination depends on the selected object selection.

By default, the coordinates have a white background and show the values based on the global origin. However, if only one object is selected and this object is of type NC-Pack or the object is located in an NC-Pack, the coordinates are displayed based on the NC-Zero Point. In this case, the coordinates are highlighted in light green.

Templates<PRO>

The Templates window allows you to import predefined elements into the current project.



tools_ball_end.vxct
tools_drill.vxct
tools_engrave.vxct
tools_flat_end.vxct

K01,5_21
K08,0_20
K06,0_30
K04,0_30
K04,0_24
K04,0_20
K04,0_12
K03,0_30
K02,0_20
K02,0_16
K02,0_10

Windows "Templates"

Here you can choose from a list of files and items to speed up your workflow and simplify project creation.

Create or modify templates

To create a new template project, either start with a new project or open an existing template and add all the desired elements to be included in the template.

Save the file in the appropriate subfolder of the **VxCraft program data folder (see chapter Installation)** with the file extension *.vxct.

Once saved as a template, the template project cannot be converted back into a regular project. Any number of objects can be created in a template, regardless of the edition. However, template projects are subject to certain functional restrictions, such as the calculation of toolpaths or postprocessor output.

Note that a template project does not store complete geometry information such as calculated jobs, models and tool geometries. Instead, only the geometry definitions required to create the geometry are stored. This allows for fast loading of the template and reduces memory requirements to a minimum. However, when opening a template, certain geometries may not be displayed initially for this reason.

Functions

Import templates
Imports the selected elements of the element list into the current project

Template file

From this list, select the file from which you want to import elements.

Elements

From this list, select the elements to be imported.

This list shows all elements of the selected file. Double-clicking on an element in the list imports this element directly into the project without having to use the "Import template" function.

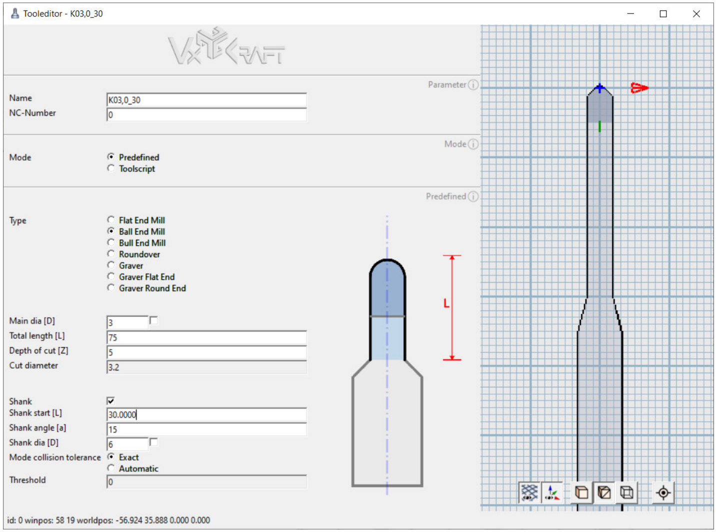
Tool editor

The tool editor allows you to create the tool geometry and define basic parameters for milling tools.

In the tool editor window, there is an area for data entry on the left side. Depending on the selected mode, two options are available. Either a script editor is available for experienced users and complex tool geometries, or a set of parameters for predefined cutter types is shown. In this case, a graphical scheme is also available to assist in entering the particular parameter selected.

On the right side of the window is the render window, which shows a two-dimensional preview of the milling tool. Here you can check the visual representation of your tool in real time and adjust the parameters to ensure that the tool has been defined correctly.

Angle specifications in the tool geometry
Unless otherwise specified (e.g. the common specification of an acute angle for engraving tools), all angles relating to the tool geometry refer to the tool axis. The values lie between 0° (parallel to the tool axis) and 90° (perpendicular to the tool axis).



Tooleditor - K03,0_30

Parameter ⓘ

Name: K03,0_30
NC-Number: 0

Mode ⓘ

Mode: ● Predefined ○ Toolscript

Predefined ⓘ

Type:
○ Flat End Mill
● Ball End Mill
○ Bull End Mill
○ Roundover
○ Graver
○ Graver Flat End
○ Graver Round End

Main dia [D]: 3 □
Total length [L]: 75
Depth of cut [Z]: 5
Cut diameter: 3.2

Shank: ☑
Shank start [L]: 30.0000
Shank angle [a]: 15
Shank dia [D]: 6 □
Mode collision tolerance: ● Exact ○ Automatic
Threshold: 0

id: 0 winpos: 58 19 worldpos: -56.924 35.888 0.000 0.000

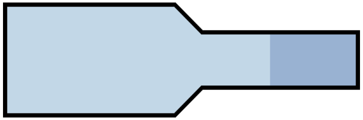
Mode "Predefined"

By selecting a predefined cutter type and setting some parameters, the tool geometry can be created quickly and easily in this mode.

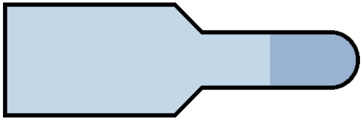
A schematic representation of the tool is located to the right of the input fields and serves as an aid by highlighting the area of the currently selected parameter. The proportions, lengths and angles in the representation may differ from the actual tool geometry.

Type

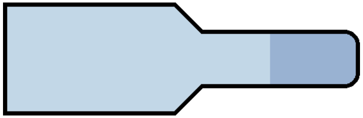
The tool type determines the cutting tip of the predefined tool.
The following tool types are available:



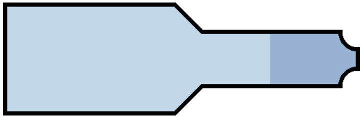
Flat End Mill
A common universal tool suitable for 3D roughing operations, for milling grooves and for complete machining of vertical surfaces and flat areas.



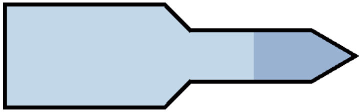
Ball End Mill
Mainly used for prefinishing and finishing 3D contours.



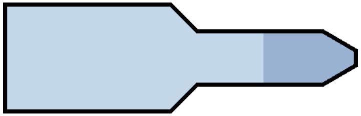
Bull End Mill <PRO>
Similar to the Ball End Mill, but with a rounded face corner for roughing and finishing operations and for machining 3D contours.



Roundover <PRO>
A form cutter for fast rounding of 2D edges. The flat face is also used as a cutting edge.

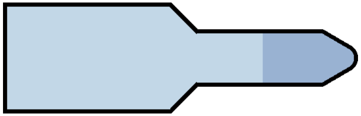


Graver
For engraving fine structures.



Graver Flat End <PRO>

A special form of graver that can also be used to finish end faces. Suitable for roughing and pre-finishing engravings.



Graver Round End <PRO>

This special form of the graver's can be used to cut off end faces and machine 3D contours. Also suitable for roughing and pre-finishing engravings.

Main diameter
Defines the nominal diameter of the milling tool.

Conicity
When enabled, this option allows the cutting geometry to be tapered so that the tool becomes narrower along its length from shank to tip.

Total length
Defines the total length of the milling tool from the tip to the end of the shank.

Depth of cut
Indicates the maximum cutting depth up to which the tool has a cutting edge.

Resulting cut diameter
This dimension is automatically calculated from the combination of depth of cut, major diameter and taper. VxCraft uses this dimension for some job calculations, among others.

Cut angle
Defines the point angle for gravers, which specifies the angle between the two cutting edges.

Diameter tip
Specifies the diameter of the flattened face for special engraving tools.

Cut radius
Specifies the radius for the rounded face cutting edge.

Shank <PRO>
If activated, a shank geometry is added to the tool.

Shank start <PRO>
Defines the starting height of the shank geometry, that is, the distance between the tool nose and the start of the shank transition.

Shank angle <PRO>
Specifies the transition angle between nominal diameter and shaft diameter.

Shaft diameter <PRO>
Defines the nominal diameter of the shank.

Shaft taper <PRO>
When enabled, this option allows the shank to be tapered so that it becomes narrower along its length from the end of the shank to the start of the shank.

Collision parameters

Mode collision tolerance
The collision tolerance mode defines how the calculation of potential collisions is handled when creating and simulating tool paths.

Due to the various options for optimizing tool paths **(see chapter "Job parameter path optimization")** even the smallest deviations of the path can lead to detected collisions. The choice of mode can influence this behavior.

Exact:
The exact mode is suitable for highly sensitive tools or if tool path optimization has been deactivated in the job. Here, any potential collision is strictly detected and reported.

Automatic:
The automatic mode calculates a tolerance threshold based on the tool geometry. Only if this threshold is exceeded, a collision is reported.

When calculating the tolerance, the resulting cutting diameter of the tool and a predefined factor are taken into account.

Tolerance threshold:
The displayed tolerance value indicates the number of pixels that may be violated by a collision during a single movement step before a collision message is issued.

Mode "Tool script"<PRO>

The tool script mode is designed for experienced users who want to create complex tool geometries using a Lua script.

This mode allows free definition of the tool geometry, but requires basic knowledge of geometry and scripting.

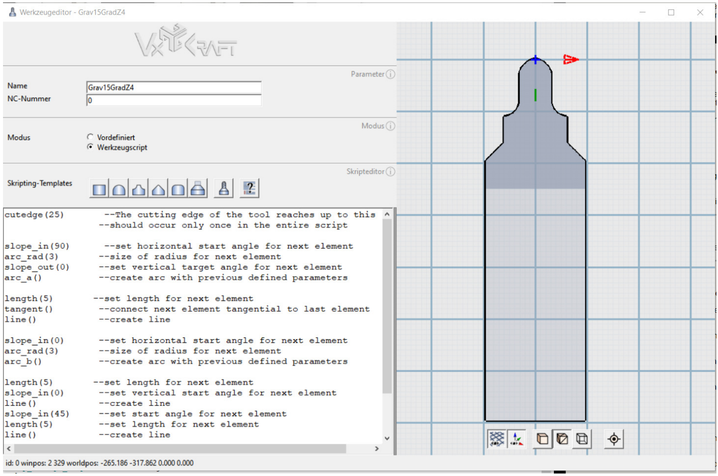
Script editor<PRO>

The script editor allows you to create a rotation profile for the tool using a series of script commands.

The commands for the **"tool script"** are described in a separate chapter of the Lua modules.

Text field

In the text field you can enter the script commands. As soon as the input focus leaves the text field or the "F5" key is pressed, the geometry of the tool is updated in the render view.



Werkzeugeditor - Grav15GradZ4

Name: Grav15GradZ4
NC-Nummer: 0

Modus: ○ Vordefiniert ● Werkzeugscript

```
cutedge(25)         --The cutting edge of the tool reaches up to this
                   --should occur only once in the entire script

slope_in(90)        --set horizontal start angle for next element
arc_rad(3)         --size of radius for next element
slope_out(0)       --set vertical target angle for next element
arc_a()            --create arc with previous defined parameters

length(5)         --set length for next element
tangent()          --connect next element tangential to last element
line()             --create line

slope_in(0)        --set horizontal start angle for next element
arc_rad(3)         --size of radius for next element
arc_b()            --create arc with previous defined parameters

length(5)         --set length for next element
slope_in(0)        --set vertical start angle for next element
line()             --create line
slope_in(45)       --set start angle for next element
length(5)          --set length for next element
line()             --create line
```

id: 0 winpos: 2 329 worldpos: -265.186 -317.862 0.000 0.000

Scripting templates

With the help of the scripting templates you can combine prefabricated script sections into a tool script or add your own script. Here is a list of all scripting templates:

Corner
Creates a horizontal line followed by a vertical line.

Rounding convex
Creates a convex 90° arc (curved outward) followed by a vertical line.

Rounding concave
Creates a concave 90° arc (curved inward) followed by a vertical line.

45° chamfer
Creates a line inclined at 45°.

Torus geometry
Creates a horizontal line followed by a vertical line with a rounded transition.

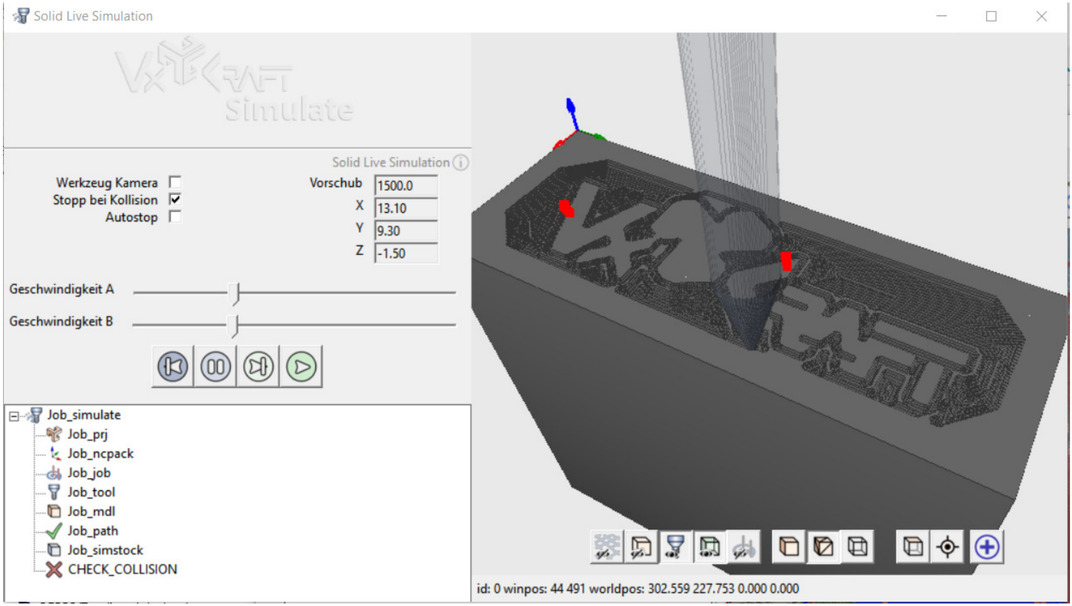
Shank
Creates an oblique line up to an absolute position in the X direction, followed by a vertical line.

Tool
Inserts a script block containing a complete tool geometry including the shank. The parameters and options for the tool can be dynamically adjusted in the upper part of the script block.

Command overview
Inserts a list of commands as a comment into the script.

Solid Live Simulation<PRO>

The calculated jobs can be visually checked using the virtual material removal simulation and analyzed for possible errors using the additional collision check.



Werkzeug Kamera □
Stopp bei Kollision ☑
Autostop □
Vorschub 1500.0
X 13.10
Y 9.30
Z -1.50

id: 0 winpos: 44 491 worldpos: 302.559 227.753 0.000 0.000

A simulation can only be carried out if both a tool and a stock are available for the job to be simulated.

The Solid Live simulation environment supports starting with multiple jobs. In this case, the desired job to be simulated can be selected in the tree structure and activated. When opening the simulation environment, the first simulation element is automatically activated.

More than one simulation window can be opened. However, running multiple simulations at the same time is strongly discouraged, as this can lead to performance degradation and instability.

> ***Tipp:***
>
> The simulation environment can also be opened with more than one job.

Simulation - collision check

In case of a collision, highlighted points are generated at the point of contact and the "CHECK_COLLISION" object is displayed. At the end of the simulation, a warning message is displayed if a collision was detected.

Simulation control

Tool camera
If the check box is activated, the camera follows the tool. The camera can still be rotated during the simulation, but it can no longer be moved.

Stop on collision
If the check box is activated, the simulation automatically stops at the collision point and issues a warning message.

> ***Note:***
>
> The collision check is always performed even if the check box is disabled.

Autostop
If the check box is activated, the simulation stops after each milling path section.

Feed rate
Displays the current feed rate.

Position
Displays the current position of the tool relative to the NC-Zero Point.

Speed A
Controls the simulation speed for rapid and plunge movements.

Speed B
Controls the simulation speed for the regular milling feedrate, excluding plunge movements.

Reset
Resets the activated simulation to the start.

Pause
Pauses the simulation.

Single step
Executes a single step. Useful for the analysis of critical areas.

Start
Starts the simulation.

Simulation tree

In this tree structure all elements for the simulation are listed.

The basic elements in the tree structure are the simulation elements. They contain all objects that are required for the execution of a simulation. They are independent of other simulation elements in the tree structure.

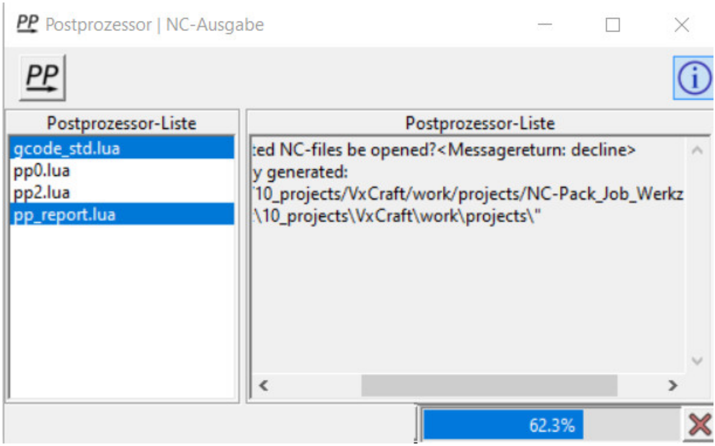
When a simulation element is activated, all associated objects are displayed and all other simulation elements are hidden. It is possible to show and hide any element manually (even from inactive simulation elements).

Simulation - Render View

The render view displays the material removal simulation. The control is identical to the render view of the main window.

Postprocessor environment

The postprocessor environment enables the conversion of NC data calculated internally by VxCraft into executable NC programs or NC documentation.



```
:ed NC-files be opened?<Messagereturn: decline>
y generated:
'10_projects/VxCraft/work/projects/NC-Pack_Job_Werkz
\10_projects\VxCraft\work\projects\"
```

Window "Postprocessor environment"

Create or modify PP scripts

In a separate **subchapter of "Lua"** you will find the script commands for postprocessors and more information about creating and customizing postprocessor scripts. By default, the PP Lua scripts are located in the **VxCraft program data folder (see the Installation chapter)**..

Postprocessor functions

Execute postprocessor
Starts the selected postprocessor run

Clicking on the "PP" icon starts the postprocessor run for the postprocessors selected in the list. The PP scripts are processed in sequence to convert the internal calculation data of VxCraft into executable NC programs or NC documentation.

Postprocessor

Select the desired postprocessors for the postprocessor run from this list.

Postprocessor console

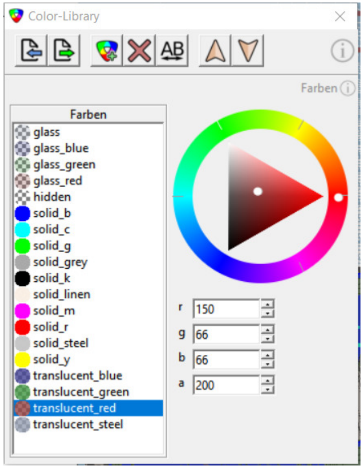
The console displays feedback from the Lua script, including information, errors, warnings, and other relevant messages.

It helps to monitor progress and possible problems during the post-processor run.

Postprocessor progress bar

The progress bar in the status bar shows the progress of the current postprocessor run. If necessary, the postprocessor run can be aborted by pressing the cancel button next to the progress bar.

Color library



Farben

- glass
- glass_blue
- glass_green
- glass_red
- hidden
- solid_b
- solid_c
- solid_g
- solid_grey
- solid_k
- solid_linen
- solid_m
- solid_r
- solid_steel
- solid_y
- translucent_blue
- translucent_green
- translucent_red
- translucent_steel

r 150
g 66
b 66
a 200

The color library is a dialog window used to manage colors for 3D objects.

It consists of a toolbar for managing the color palette, a list of existing colors and an area for selecting and setting a color.

The color palette is loaded from the system file colors.lua **(see also chapter Lua/Module Settings)** and is stored independently of the current project. Changing the color palette therefore affects both future and existing projects.

Changes in the color palette are saved immediately. For this reason, when opening the dialog window, you will be asked whether a backup of the current color palette should be created in the settings directory of VxCraft.

Functions

The following functions are available for managing the color palette:

Import colors
Opens the file manager to select a color file and imports the colors from it.

You will be asked whether the colors should be added to the existing palette or whether the palette should be emptied beforehand.

Export colors
Opens the file manager to select a destination file and exports the current color palette.

Add color
Adds a new color to the palette.

Delete color
Removes the selected color from the palette.

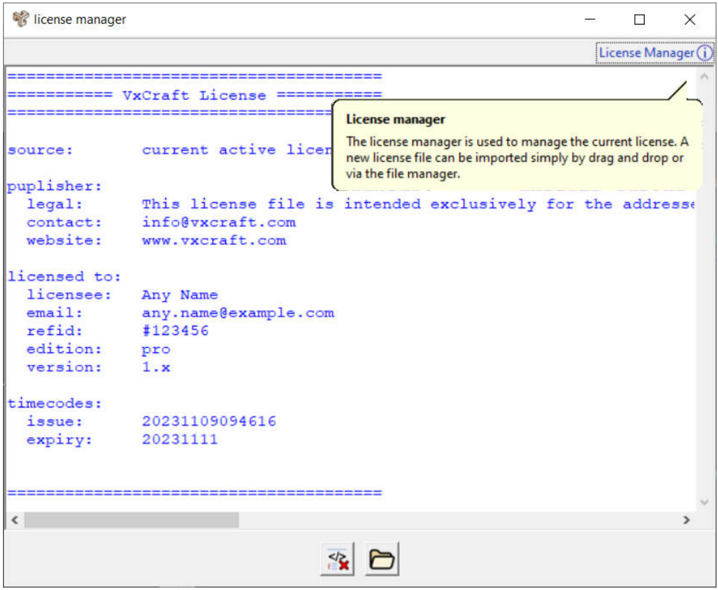
Rename color
Renames the selected color.

Move color up
Moves the selected color up in the list.

Move color down
Moves the selected color down in the list.

License manager

The license manager is used to manage the current license. A new license file can be imported simply by drag and drop or via the file manager.



```
=======================================
=========== VxCraft License ===========
=======================================

source:        current active licen

puplisher:
  legal:       This license file is intended exclusively for the addresse
  contact:     info@vxcraft.com
  website:     www.vxcraft.com

licensed to:
  licensee:    Any Name
  email:       any.name@example.com
  refid:       #123456
  edition:     pro
  version:     1.x

timecodes:
  issue:       20231109094616
  expiry:      20231111


=======================================
```

License manager

The license manager is used to manage the current license. A new license file can be imported simply by drag and drop or via the file manager.

Text box

When the License Manager is opened, the license loaded at **program startup** is displayed in the text box.

When a license is loaded or reset, information about the status of the new license is displayed in the text window.

Font color

If the font in the text box is red, there is a license problem. More information about the error is displayed in the text box.

Blue means that a purchased license is active or has been successfully imported into the system.

Important: A successfully imported license does not mean that it is valid, the license manager only performs a Basic check.

Reset license

Removes a purchased license from the system and activates the Free Edition after a restart of VxCraft.

After a time-limited license has expired, this feature can be used to remove the license to activate the Free Edition.

This feature is only activ if a purchased license is installed in addition to the Free Edition license (available with every installation).

Import license

Opens the file manager to import a purchased license. VxCraft must be restarted to fully activate a license.

Import license via drag & drop

In addition to the file manager, a license file can be easily dragged and dropped into the license manager, e.g. directly from the attachment of the received e-mail.

Objects

This chapter describes all object types of the project tree and the corresponding configuration dialogs. An overview of the hierarchical project structure can be found in the chapter **Project tree**.

General functions

This section describes general functions for objects. The availability of a function depends on the object type.

Export
Opens File Explorer to select a destination file and exports the object in the selected file format.

This function allows you to save the geometry or data of the selected object in a suitable file format for further use in other applications or projects.

Refresh
Performs an update of the geometry or data of the object if it depends on references and these references have changed since the last geometry creation or data creation.

Refreshing ensures that the selected object takes into account the latest information from the references and is up to date.

Undo / Redo

Controls the steps back or forward for the object to undo or redo changes to geometry, data, or other settings.

- **Undo:** Undoes the last change made to the selected object and returns to the state that existed immediately before the change.
- **Redo:** Restores the last undone change to the selected object and returns to the state that existed immediately after the change.

Please note that the "Undo" and "Redo" functions only affect the current object and do not affect other objects or the entire scene.

Project settings

In the project settings you can adjust various parameters for the current project. Note that the changes made here are valid only for the current project and have no effect on other projects.

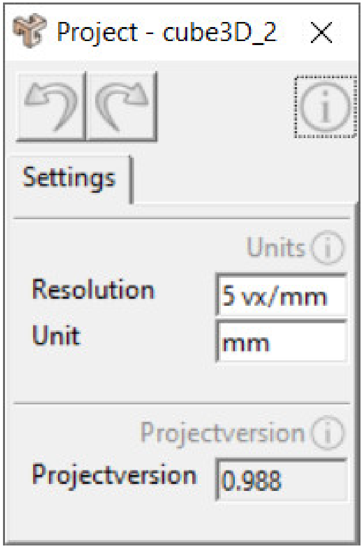
Units

In this section you can set the project resolution and the unit for the current project.

Voxel
A voxel is the basic invariant unit of VxCraft and can be considered as a three-dimensional pixel. The toolpath calculation uses the voxel grid. All geometries are interpolated into the voxel grid before calculation, losing details smaller than one voxel.

> ***Note:***
> The name of the preset user unit can be changed in the settings.

User unit
Determines the name of the user unit. The name has no influence on the dimensions in VxCraft. It is only used to distinguish between voxel and user unit and can be chosen at will. However, it is recommended to choose the unit as it is used on the CNC machine.



Resolution: 5 vx/mm
Unit: mm
Projectversion: 0.988

© vxcraft.com

Project configuration dialog

Project resolution *[ppVxPerUnit] [ppUnitPerVx]*
The project resolution determines the number of voxels per user unit. It defines the level of detail of the calculated toolpath.

> ***Tip:***
>
> A suitable project resolution can make the workflow much easier. It is recommended to adapt the resolution to the graphics program used. In most graphics programs, print resolutions can be specified in "mm" in addition to "inch". This way, length measurements can be used in the graphics program that can later be traced in VxCraft.

Note that a high resolution leads to a very dense voxel grid. This can lead to longer calculation times and reduce the maximum model size (user units). In the following example table you will find guideline values that should simplify the handling of the project resolution:

User unit	Project resolution	Accuracy	Model size (approx.)
mm	1	1mm	approx. 3.000mm
mm	5	0.2mm	approx. 600mm
mm	10	0.1mm	approx. 300mm
inch	20	0.05 inch	approx. 150 inch
inch	100	0.01 inch	approx. 30 inch
inch	300	0.0033 inch	approx. 10 inch

In the **chapter "Render Window"** the units are explained in more detail in relation to the workplane of the render window.

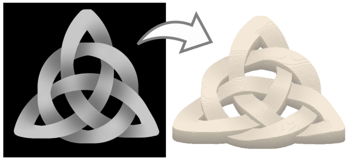
Project version

Shows the VxCraft version under which the project was last saved.

Heightmap model

A heightmap model is a special type of 3D model where the geometry is defined by a two-dimensional graphic with additional height information.

In contrast to conventional 3D models, the design of the object is based solely on the surface, while side and bottom surfaces are derived from this surface and thus have no independent geometry.



Heightmap aus einer Grafik

© vxcraft.com

Geometry mode

A heightmap model can be created, loaded or generated by various methods.

In the following section all available modes are explained:

Graphic

The heightmap model is created from an imported graphic, with the color value of each pixel determining the height. High color values create raised geometry. A color value of 0 creates a break in the model. The following tools and settings are available:

Import graphic

Opens the file manager to import a graphic. The size of the heightmap in the unit "voxel" corresponds to the resolution of the imported graphic. The **chapter "System"** explains the available file formats in more detail.

Reload graphic

Re-reads the already imported graphic from the saved file path and updates the heightmap model.

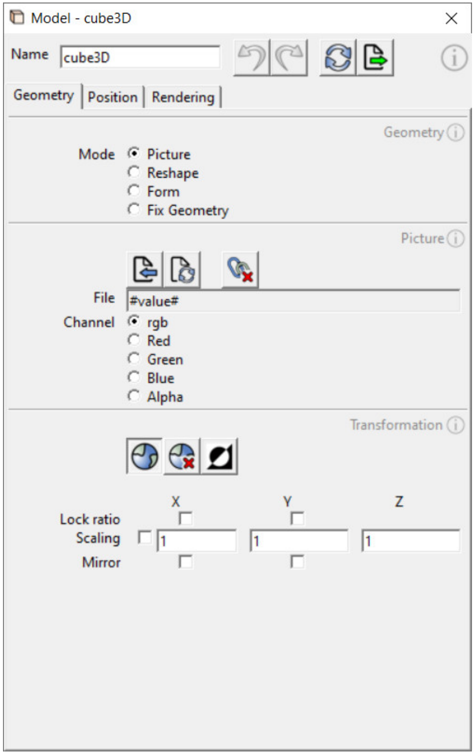
Detach file path

This function removes the linked file path from the object, but keeps the data of the last loaded file.

Color channel filter

Enables the use of specific color channels of the graphic.

- **rbg:** Calculates the average brightness value of all color channels.
- **Red:** Uses only the red color channel for height determination.
- **Green:** Uses only the green color channel for height determination.
- **Blue:** Uses only the blue color channel for height determination.
- **Alpha:** Uses the alpha channel to determine the height.



Name: cube3D

Mode: ● Picture ○ Reshape ○ Form ○ Fix Geometry

File: #value#

Channel: ● rgb ○ Red ○ Green ○ Blue ○ Alpha

Lock ratio: X □ Y □

Scaling: □ X 1 Y 1 Z 1

Mirror: X □ Y □

Configuration dialog

Reshape

The heightmap model is formed from one or more existing models.

Shape

Determines the resulting shape.

- **Heightmap** The height information for the heightmap is calculated using the geometry references and the selected orientation reference.
- **Box** The referenced geometry is enclosed in a rectangular box aligned with the orientation reference - useful for creating a stock.

Allowance

An allowance can be defined individually for each axis direction (starting from the orientation reference). Positive values expand the geometry in the respective direction, while negative values shrink it.

Geometry reference

One or more models can be selected to be merged into a single heightmap.

> ***Note***
>
> In this mode, the position and orientation of the object are fixed by the selected references and cannot be changed directly. In order to be able to edit the position, the geometry created from the references can be fixed using the "Fix geometry" mode.
>
> ***Tip***
>
> Tool models can also be converted into a heightmap.

Orientation

The orientation of the resulting heightmap is aligned with the selected reference. The height axis of the heightmap corresponds to the Z axis of the selected reference.

Create 3D cast <PRO>

The "Invert" transformation tool can be combined with "Reshape". This creates an impression of the selected geometry references.

Shape

Creates a heightmap model based on a basic geometric shape and user-defined dimensions.

Basic geometric shapes

- **Block:** Rectangular block
- **Cylinder:** cylinder aligned along the Z-axis

Dimensions

- **Size X:** Model size in X direction
- **Size Y:** Model size in Y direction
- **Size Z:** Model size in Z direction
- **Frame thickness [optional]:** The specified dimension determines the frame thickness of the model. If no dimension is specified (0.0), the model is created as a solid shape.

Fix geometry

Detaches the model from its geometry and position references.

When the "Fix geometry" mode is activated, the geometry and the position of the model are fixed. The dependency on geometry and position references is removed, making the model independent and allowing it to be positioned freely.

Heightmap Transformation

The transformation changes the underlying 2D dataset of the heightmap from which the 3D geometry is generated.

The position and rotation of the object are not affected by this. The transformation is applied after the 2D dataset has been read in and before the 3D geometry is generated and the model is positioned and rotated by the settings in the "Position" tab. The parameters for the transformation are also retained when the 2D base dataset is updated (e.g. by reloading a graphic).
A common transformation that is often applied is the scaling of a heightmap in the direction of the Z-axis. Since the height of a heightmap often needs to be adjusted after importing the graphic, this type of transformation is particularly relevant.

Transformation tool
Enables/disables the transformation for the object.

Reset transformation
Resets the transformation to the default values.

Invert heightmap
Inverts the height information of the individual channels of the graph.

Lock ratio
Locks the scaling ratio for the X and Y axes to preserve the proportions of the model.

Scaling
Scales the heightmap to the desired size. The scaling mode can be switched between "scaling factor" and "absolute scaling" with the checkbox.

Mirror
Mirrors the geometry along the specified axis.

3D model<PRO>

A 3D model is a three-dimensional representation of an object composed of triangles.

A 3D model can have any level of detail and size. During NC calculation, the model is transferred to the voxel grid, which reduces the level of detail depending on the project resolution. Unlike heightmap models, which define their geometry by a 2D graphic with height information, 3D models have a complete geometry with all degrees of freedom.

© vxcraft.com

Geometry mode

A 3D model can be created, loaded or generated by various methods. In the following section all available modes are explained:

File

The 3D model is created from an imported STL file. The following tools and settings are available:

Import 3D model
Opens the file manager to import a 3D file. The **chapter "System"** explains the file formats in more detail.

Reload 3D model data
Re-reads the already imported file from the saved file path and updates the 3D model.

Detach file path
This function removes the linked file path from the object, but keeps the data of the last loaded file.

Reshape

The 3D model is formed from one or more existing models.

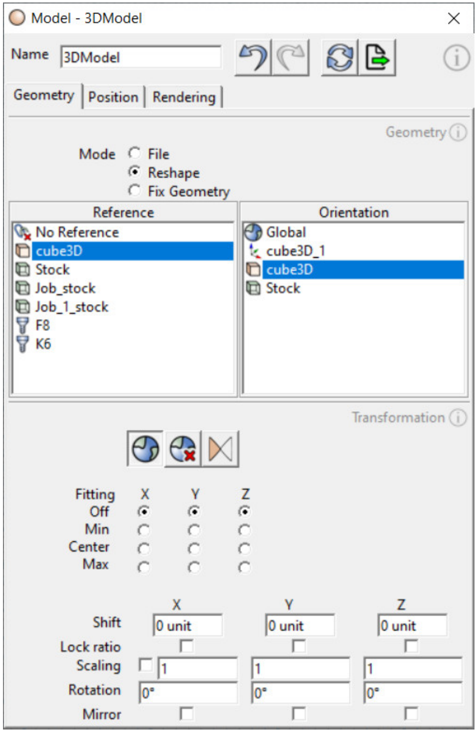
> ***Hinweis***
>
> In this mode, the position and orientation of the object are fixed by the selected references and cannot be changed directly. In order to be able to edit the position, the geometry created from the references can be fixed using the "Fix geometry" mode.
>
> ***Tip***
>
> Tool models can also be converted into a 3D model.

Geometry reference

One or more models can be selected to be merged into a single 3D model.

Orientation

The orientation of the resulting 3D model is aligned according to the selected reference. The position and orientation of the model are determined by the selected reference.



Name: 3DModel

Mode: ○ File ● Reshape ○ Fix Geometry

Reference	Orientation
No Reference	Global
cube3D	cube3D_1
Stock	cube3D
Job_stock	Stock
Job_1_stock	
F8	
K6	

Fitting	X	Y	Z
Off	●	●	●
Min	○	○	○
Center	○	○	○
Max	○	○	○

	X	Y	Z
Shift	0 unit	0 unit	0 unit
Lock ratio	□	□	□
Scaling □	1	1	1
Rotation	0°	0°	0°
Mirror	□	□	□

3D model - configuration dialog

Fix geometry

Detaches the model from its geometry and position references.

When the "Fix Geometry" mode is activated, the geometry and the position of the model are fixed. The dependency on geometry and position references is removed, making the model independent and allowing it to be positioned freely.

Lock geometry data

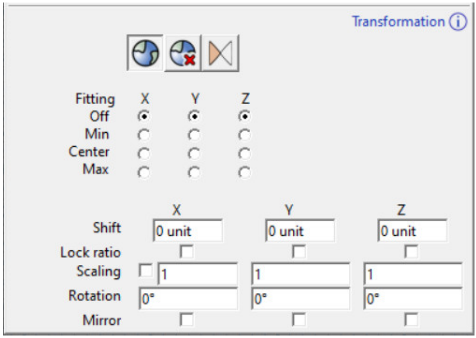
Locks the geometry data of a 3D model to protect the data from unwanted readout (export lock)

If a lock is set, it cannot be released again. A model with locked geometry data cannot be exported and also the geometry mode cannot be changed anymore. In addition, it cannot be selected by another 3D model as a geometry reference. Nevertheless, it is possible to apply a transformation to a locked geometry, and the object can still be positioned freely.

3D model transformation

The transformation changes the position of the model origin and the geometry of the 3D model relative to the model origin.

This function allows you to adjust the shape and size of the model without affecting its position in 3D space. It is important to distinguish the transformation from the positioning of the model: While the transformation affects the geometry of the model and, if necessary, moves the model origin, the settings in the "Position" tab change the location of the model origin in 3D space.



Fitting	X	Y	Z
Off	●	●	●
Min	○	○	○
Center	○	○	○
Max	○	○	○

		X	Y	Z
Shift		0 unit	0 unit	0 unit
Lock ratio		□	□	□
Scaling	□	1	1	1
Rotation		0°	0°	0°
Mirror		□	□	□

A commonly used transformation is the scaling of the 3D model to bring it to a specific dimension.

Transformation Tool
Enables/disables the transformation for the object.

Reset transformation
Resets the transformation to the default values.

Invert surfaces
Inverts the surfaces of the 3D model.

This feature can be useful when an imported model has the surface normals misaligned, causing the inside to be rendered instead of the outside. Inverting the faces changes the orientation of the normals so that the inside of the faces faces out and vice versa. This function only has a visual effect and does not affect the geometry of the model.

Fitting
Customizes the origin of the 3D model along the X, Y, and Z axes based on the size of the model geometry.
Four options are available for each axis to set the origin: Off, Min, Center and Max.

- **Off:** No change in model origin.
- **Min:** Sets the origin of the model to the position of the minimum extension of the geometry along the axis.
- **Center:** Positions the origin of the model at the center of the geometry along the axis.
- **Max:** Sets the origin of the model to the position of the maximum extension of the geometry along the axis.

This function allows you to precisely adjust the origin of the model, for example to align it on a specific plane or axis, depending on the size and extent of the model geometry. The "Move" function is deactivated for the respective axis when the origin of the model is set by "Fitting".

Shift
Moves the origin of the model in the X, Y and Z axes.

Lock ratio
Locks the scaling ratio for the X, Y and Z axes to preserve the proportions of the model.

Scaling
Scales the 3D model to the desired size. The scaling mode can be switched between "scaling factor" and "absolute scaling" with the checkbox.

Rotation
Rotates the 3D model by the specified number of degrees in the X, Y and Z axes.

Mirror
Mirrors the geometry along the specified axis.

Tool

The geometry and parameters for the milling tool are defined in the tool object. The geometry can be defined in the separate tool editor. VxCraft uses the geometry for radius correction and collision checking.

Tab "Definition"

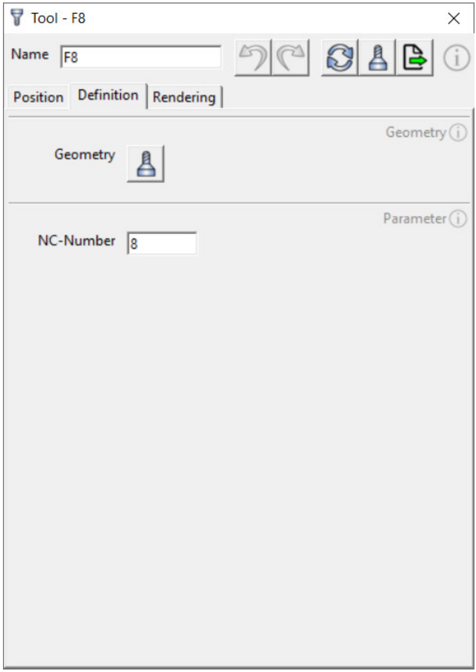
Geometry

The tool editor is available for creating the geometry. The tool editor is described in the chapter for **special dialogs**.

Parameter

In this section geometry incompatible parameters for the tool are listed

NC number *[ppToolId]*
The NC number is intended for the tool changer and must match the tool list of the machine control. The postprocessor writes the tool change command with this number before the job starts.



Name: F8

NC-Number: 8

Tool configuration dialog

NC-Pack

An NC-Pack bundles a group of jobs and stocks and defines the NC-Zero Point.

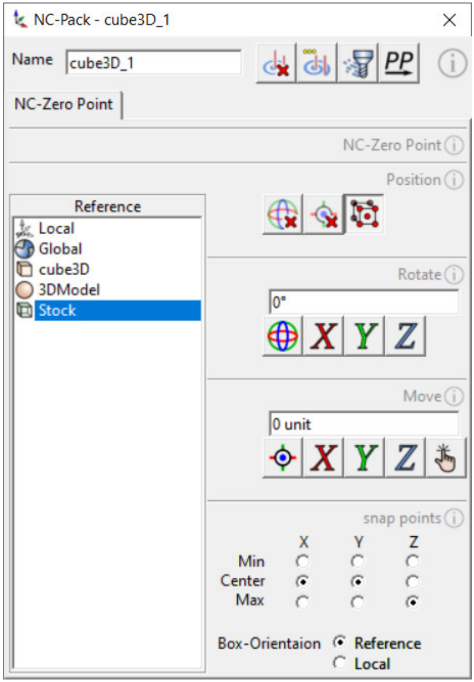
NC-Zero Point

The NC-Zero Point determines the orientation of the stock in the machine for tool path calculation.

It is displayed as a coordinate system in the rendering window and can be defined by positioning the "NC-Pack" object. The NC-Zero Point is applied to all jobs that are in the NC-Pack.

Multi-sided machining <PRO>

With the ability to tilt the NC-Zero Point about any axis, models can be machined from all sides.



Name: cube3D_1

Reference:
- Local
- Global
- cube3D
- 3DModel
- Stock (selected)

Rotate: 0°

Move: 0 unit

	X	Y	Z
Min	○	○	○
Center	●	●	○
Max	○	○	●

Box-Orientaion ● Reference
○ Local

NC-Pack - configuration dialog

Stock

A stock is essentially a heightmap model that has the same functions for generation and positioning.

Further details about these functions can be found in the **chapter "Heightmaps"**.

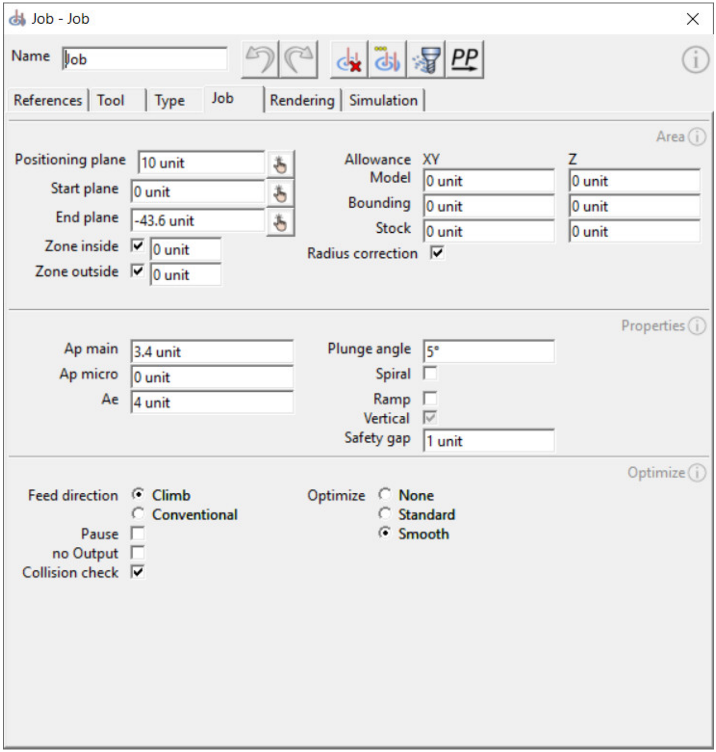
In the project hierarchy, stocks are on the same level as jobs. Stocks from jobs can be selected as references, provided they are located above the respective job in the hierarchy.

Job

A job object contains all references and parameters needed for the calculation of a toolpath. After successful calculation, this object contains the resulting toolpath.

> ***Note:***
>
> The NC-Zero Point for the job is defined by the NC-Pack.

The job must be recalculated if a parameter changes or if a referenced object has changed.



Name: Job

Positioning plane: 10 unit
Start plane: 0 unit
End plane: -43.6 unit
Zone inside ☑ 0 unit
Zone outside ☑ 0 unit

Allowance	XY	Z
Model	0 unit	0 unit
Bounding	0 unit	0 unit
Stock	0 unit	0 unit

Radius correction ☑

Ap main: 3.4 unit
Ap micro: 0 unit
Ae: 4 unit

Plunge angle: 5°
Spiral □
Ramp □
Vertical ☑
Safety gap: 1 unit

Feed direction: ● Climb ○ Conventional
Pause □
no Output □
Collision check ☑

Optimize: ○ None ○ Standard ● Smooth

Configuration dialog

Geometry references

Geometry references determine the geometry of the object to be manufactured. For a reference to be available to the job, it must be above the job in the project hierarchy.

Multiple selection is also allowed for each reference type.The following is a listing of all geometry references:

Model reference
Model references determine the shape of the object to be milled. The milling tool moves along the contour of the models.

In general, the tool paths are calculated so that the tool does not exceed the model contour. However, you can set a negative allowance and deactivate the radius compensation of the tool. In such cases, the toolpath either lies directly at the model contour or even exceeds it.

Stock reference
The stock (blank) determines the area to be machined. Toolpaths are usually generated only within the stock.

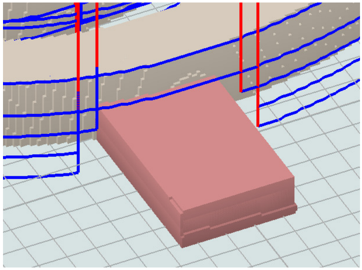
An exception are approach paths and separate settings in the job parameters, such as zone limits and allowance options. If no stock is selected, a workpiece is emulated internally in the size of the selected model references and all areas are processed. However, for some Job strategies, a workpiece reference may be mandatory. A workpiece reference is only available to the job if the object is above the job in the project hierarchy.

Bounding reference <PRO>

Defines areas that are excluded from processing.

Similar to the model reference, the toolpaths do not exceed this area or only according to the exceptions and rules already described in the model reference. The difference between a model reference and bounding model reference lies in the contour machining: The contour of a bounding model is not machined. If the toolpath encounters a bounding model, it ends at this point, lifts off and continues machining in the next area to be machined.

Bounding model references can be used to omit areas within a processing area, such as holding bars. A bounding model is only available to the job if the object is above the job in the project hierarchy.


© vxcraft.com

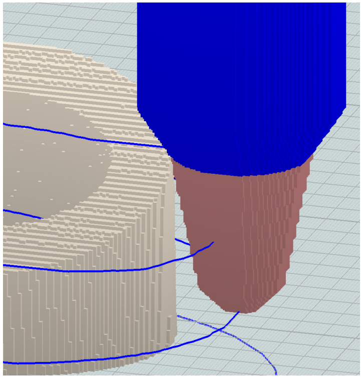
Example holding bar as bounding reference

Tool Reference

The tool reference determines the radius compensation, is used by the collision check and contains further parameters for the tool path calculation.

A tool reference is essential in most cases, because without a tool neither a radius correction nor a collision check can be performed. The tool referenced in VxCraft must match the tool actually used on the machine. Otherwise, the milling result may be falsified and there is a risk of collisions during the milling operation.

It is important that all relevant tool parameters, first and foremost the geometry definition, are correctly defined in the tool.


© vxcraft.com

Radius correction

Job specific tool parameters

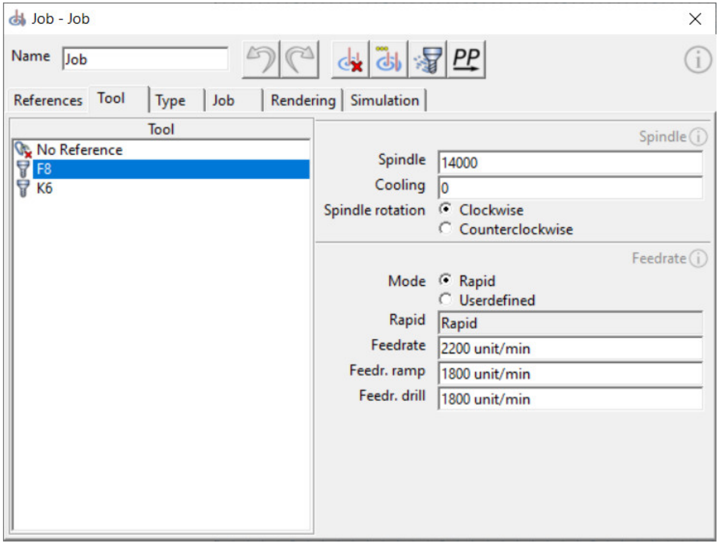
These parameters define the tool and spindle settings for the current job.

Any number of job objects can reference the same tool, but the job-specific tool parameters apply only to the job in question. This allows the cutting data to be optimized and individualized for each machining step.

Spindle speed *[ppSpindlespeed]*
This parameter defines the spindle speed.

Cooling *[ppCooling]*
Defines the cooling mode for processing. The value depends on the postprocessor used.

Direction of spindle rotation *[ppSpindleRotation]*
Determines the direction of rotation of the spindle (clockwise or counterclockwise).



Name: Job

Tool
- No Reference
- F8
- K6

Spindle: 14000
Cooling: 0
Spindle rotation: ● Clockwise ○ Counterclockwise

Mode: ● Rapid ○ Userdefined
Rapid: Rapid
Feedrate: 2200 unit/min
Feedr. ramp: 1800 unit/min
Feedr. drill: 1800 unit/min

Tool job parameters

Rapid traverse mode *[ppFeedMode]*
Controls the feed rate for rapid traverse movements. The rapid can be controlled either by the CNC control (Rapid) or by an optional value input (User defined) in the job.

Feed rate rapid *[ppFeedRapid]*
Optional feed rate for positioning movements in unit/min. Only active if rapid traverse mode is set to "User defined".

Feed rate *[ppFeed]*
The feed rate for cutting movements in unit/min.

Feed rate ramp *[ppFeedRamp]*
The feed rate for ramped plunge movements into the material in unit/min.

Drill feed *[ppFeedDrill]*
The feed rate for vertical plunge movements into the material in unit/min.

Type*[ppJobtype]*

Selecting the Job strategie determines the basic calculation algorithm for the toolpath.

Roughing *[ppJobtype=0]*
Clears the 3D model from the stock layer by layer.

This algorithm creates planar tool paths on each infeed plane for broaching the stock. Between the start and floor planes, planes are created and broached at the distance of the main infeed (Ap Main). After each main infeed, the residual material is optionally broached layer by layer from bottom to top with a finer layer gradation (Ap Micro). As a result, the 3D model is machined out very quickly and thoroughly during roughing.

Contour *[ppJobtype=1]*
The contour milling algorithm creates plane-by-plane, planar toolpaths around the 3D model. With a low, vertical cutting depth (Ap Main), this Job strategie is also suitable for finishing steep and vertical areas.

3xShape <PRO> *[ppJobtype=2]*
With the "3xShape" finishing algorithm, flat areas can be three-dimensionally mapped. The spiral cutting depth reduces approach and departure movements to a minimum.

Engrave *[ppJobtype=3]*
The engrave algorithm is optimized for fine structures that are "embossed" in one plane.

The structures are machined out in a spiral using three-dimensional tool paths. The vertical cutting depth adapts to the model. If the depth of the structure exceeds the main vertical infeed, the plane is completely cleared after each main vertical infeed before machining is continued in depth.

Drill <PRO> *[ppJobtype=4]*
The "Drill" Job strategie is a semi-automatic cycle for calculating drilling movements.

Parameter

In the "Parameters" tab, the settings for the selected Job strategie can be adjusted. This section varies greatly between Job strategies, as not all parameters are relevant or available for every Job strategie.

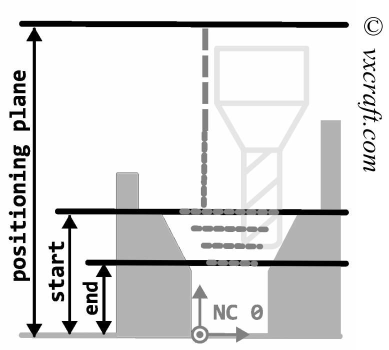
Parameters - Milling area

In this section, the milling area to be calculated is defined and settings for the geometry references are made.

Vertical milling area

The following levels limit the vertical milling area and determine the positioning plane for the job.

Positioning level *[ppPlaneSite]*
The positioning plane defines the height of the plane relative to the NC-Zero Point. To avoid collisions, this plane must be higher than the geometry references. At the height of the positioning plane, the tool usually moves to the various machining positions at rapid speed. Whether this plane is approached before start positioning and after a tool change depends on the postprocessor.



Start plane (top machining plane) *[ppPlaneStart]*
The start plane determines the height of the plane where machining starts, relative to the NC-Zero Point. This plane limits the vertical milling area upwards. The milling paths are generated from this plane. Plunge movements, which can also start above, are an exception.

End plane (lowest working plane) *[ppPlaneEnd]*
The end plane sets the height of the plane where machining ends, relative to the NC-Zero Point.

This plane limits the vertical milling area downwards. The milling paths are generated up to this plane.

Zone limitation

The milling area can be restricted by zone settings in addition to the selected geometry references.

Zone inside/outside <PRO>

The zone boundary runs along the outermost silhouette of the model references. Optionally, the zone can be extended or reduced with an offset. Zone limitation can be very useful when using holding bars and can even reduce the number of necessary geometry references.

Zone inside <PRO> *[ppAreaIn]*

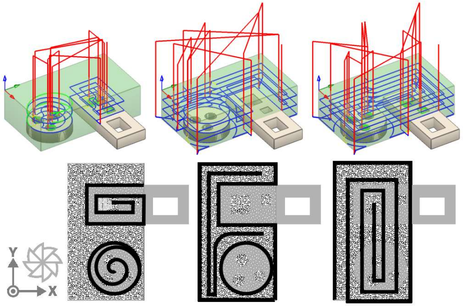
Toolpaths are created within the outermost model silhouette. By specifying an offset, the zone can be extended or reduced. Islands and holes in the geometry references are not affected and will be processed.

Zone outside <PRO> *[ppAreaOut]*

Toolpaths are created outside the outermost model silhouette. The zone can be extended or reduced by specifying an offset.

Zone stock <PRO> *[ppAreaStk]*

Tool paths are bounded to the area of the stock. This option is available for Job strategies where it is not common to limit the toolpath to the stock only.



© vxcraft.com

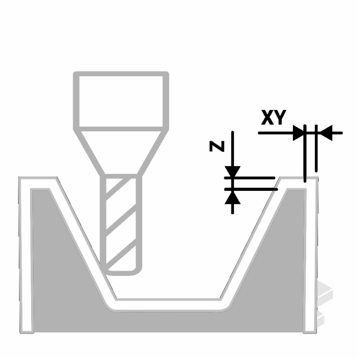
Zone limitation
Zone Inside | Zone Outside | Both zones
Zone stock: activated

Geometry allowance

<PRO>

[ppAll...]

In the allowance table, a material allowance can be specified in the horizontal axis of the NC-Zero Point (XY) as well as the vertical axis (Z) for each geometry reference. Negative values are also possible.



© vxcraft.com

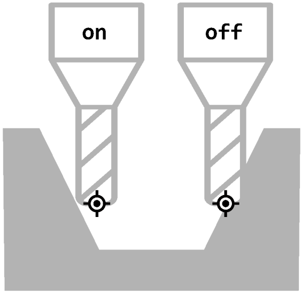
3D Radius correction

[ppRadcorr]

The radius correction takes the tool geometry into account when calculating the tool paths and enables a collision check to be performed.

With radius compensation activated, the collision check is applied and the tool path is calculated so that the tool geometry does not penetrate the model contour, or only according to the tolerance and geometry allowance.

If the radius compensation is not activated, the toolpaths are applied directly to the model geometry. In this situation, the inserted (unknown) tool would enter the model geometry up to half of the diameter or up to the tool axis. A collision check is not possible in this case.

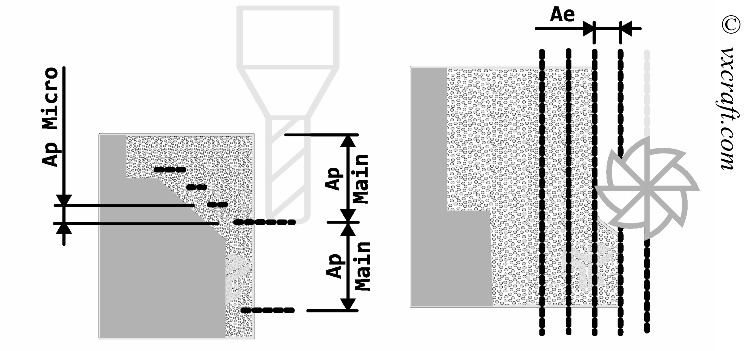


© vxcraft.com

Parameters - properties

The algorithm of the selected Job strategie is controlled in the job properties.
Among other things, the cutting depth, approach and departure movements, plunging behavior and Job strategie-specific properties are configured here. This area varies greatly between the different Job strategies.

Cutting depth



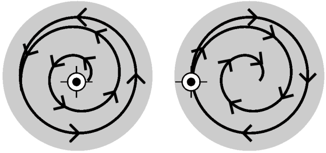
Vertical main cutting depth *[ppApMain]*
Starting from the start plane (uppermost machining plane), the main cutting depth is fed in the direction of the Z axis until the end plane (lowermost machining plane) is reached.

Vertical cutting depth of the intermediate steps <PRO>
[ppApMicro]
The material that cannot be reached by the main cutting depth is worked out in the distance of the intermediate steps.

The main cutting depth does the roughing work, while the intermediate steps clean the 3D contour by a finer gradation. The value is optional and must be smaller than the main infeed.

Horizontal cutting depth *[ppAe]*
Specifies the preferred horizontal spacing of the milling paths.
Regardless of this value, cuts with higher distances or even cuts into the full material can also occur in the calculated toolpath due to the geometry. The wrap angle (pressure angle) of the milling cutter can therefore be larger than would be expected from the specified value.

Direction



"Start Center" and "Start Outside in climb.

Start
Defines the starting point of the milling path.

Center
The milling path starts in the center and ends at the edge of the area to be machined.

Outside
The milling path starts at the edge of the area to be machined and ends in the center.

Plunge / lateral approach

In order to reach the actual starting point of the milling operation, an attempt is made, if possible, to avoid plunge movements and instead to prefer lateral approach movements. In a lateral approach, the tool is positioned either on an area that has already been machined or laterally outside the stock, and from there it advances horizontally to the starting point of the actual milling operation.

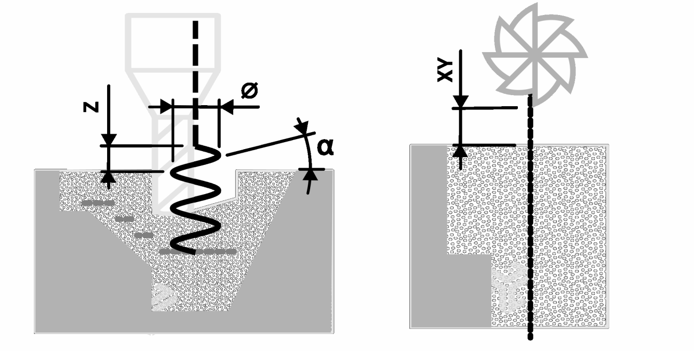
If a lateral approach is not possible, plunge movements in the direction of the Z axis are used according to a priority sequence. The sequence from high to low is: spiral plunge, ramp plunge and finally vertical plunge.

The available plunge variants depend on the selected Job strategie, and the individual plunge options can be activated or deactivated as required.

Plunge angle α *[ppPlungeAngle]*
Specifies the angle for spiral and ramp plunge movements.

Spiral plunge <PRO> *[ppPlungeSpiral]*
The tool plunges into the stock in a spiral from the stock surface to the working plane.

The dimension of the spiral is determined by specifying the diameter *[ppPlungeDia]* and the plunge angle, while the height is calculated from the working plane, the workpiece surface and the safety clearance in the Z axis.



Plunge parameter and safety distance

© vxcraft.com

Ramp plunge <PRO> *[ppPlungeRamp]*
The tool plunges into the stock in a ramp-like manner.

The starting height of the ramp is calculated from the machining plane, the workpiece surface and the safety clearance. In the XY direction, the path of the ramp runs along the toolpath immediately following the working plane. If a longer path is required for the ramp than is available, the path repeats itself in a pendulum shape until the machining plane is reached.

Vertical plunge *[ppPlungeDrill]*
The approach movement is vertical through the stock, using the drill feed.

Vertical plunge has the lowest priority and is only used if none of the other plunge variants are possible or if they are deactivated.

Safety distance *[ppSgap]*
The safety clearance defines the minimum distance in all three axis directions (X, Y and Z) between the tool and the geometry references for positioning movements in rapid.

In rapid traverse, the tool approaches the geometry up to the set-up clearance. Ramped plunge movements are extended up to this distance.

> ***Tip:***
>
> A large value for the safety distance significantly influences the processing and calculation time and also extends the milling time due to longer plunge paths. The value should be selected in such a way that safe milling is ensured, taking into account all tolerances and deviations, without being unnecessarily large.

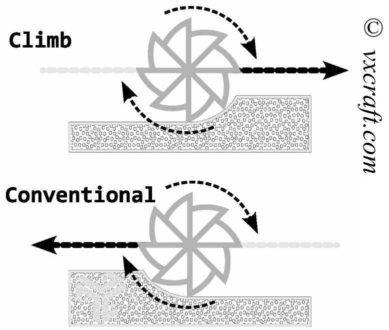
Preferred running direction *[ppFeedDir]*
The preferred running direction determines the direction of movement of the tool in relation to the model contour.

It should be selected taking into account various factors, such as material, machine, cutting tool and its tool life, desired surface finish, accuracy and infeed strategy. A detailed description of all influencing factors would go beyond the scope of this manual. For some applications, such as a full cut, this parameter does not play a role.

As already mentioned in the heading, this parameter is the **preferred** running direction. This means that an attempt is made to use the running direction set here. However, depending on the geometry and other job parameters, this may change several times in the resulting toolpaths.

Climb *[ppFeedDir=0]*
In climb milling, the tool works its way forward in the feed direction.

The cutting edges of the tool hit the material abruptly, whereby the chip thickness is maximum at the beginning and decreases steadily until the cutting edges exit. In practice, the climb is usually selected for CNC milling nowadays, provided that the guide clearance of the machine allows it. This method generally causes less vibration and less tool wear and usually produces a better surface quality on the stock.



Conventional *[ppFeedDir=1]*
In conventional milling, the tool moves in the opposite direction to its direction of rotation.

The chip forms away from zero as the cutting edge enters, towards the maximum chip thickness as the cutting edge exits. The conventional can be advantageous when machining harder material or roughing, as it allows a higher metal removal rate and improves chip formation. However, the conventional can lead to increased tool wear and vibration, possibly affecting surface quality and tool life.

Pause *[ppPause]*

Inserts a pause command into the NC program.

The position at which the pause is actually set depends on the postprocessor. Usually this is done after the tool change, before the approach of the first position and before the start of the spindle.

No NC output
If this parameter is enabled, the job will not be forwarded to the postprocessor.

Dhis can be useful, for example, if an object that has already been milled is to be reproduced as a virtual blank for further machining.

Collision check
If the check box is activated, a collision check is performed for the calculated toolpath.
Attention: A deactivated collision check can represent a risk for the machine and safety!
For more information, see the subchapter **"Collision report"**.

Path optimization

[ppOpt]

By selecting a path optimization, the calculated milling path is subsequently improved.

No path optimization *[ppOpt=0]*
The calculated raw path is not optimized.The generated path provides the highest possible precision, but results in very fine staircase-like movements. This mode can be useful if the CNC machine control is to smooth the path.

Standard *[ppOpt=1]*
Staircase-like movements are reduced by smoothing the calculated raw path. Due to a low tolerance, the details of the model are preserved as far as possible.

This option ensures an optimal balance between precision and machine performance.

> ***Note***
>
> Path optimization can possibly lead to collisions. A mode for the tolerance threshold can be set in the tool editor. This threshold must be exceeded before a collision is reported.

Smooth *[ppOpt=2]*
The calculated raw path is generously smoothed and rounded. This results in very smooth tool movements that are gentle on the material and allow the use of high feed rates.

However, since the tolerance is higher in this case, the toolpath can deviate further from the original geometry, which can result in the smallest details being affected.

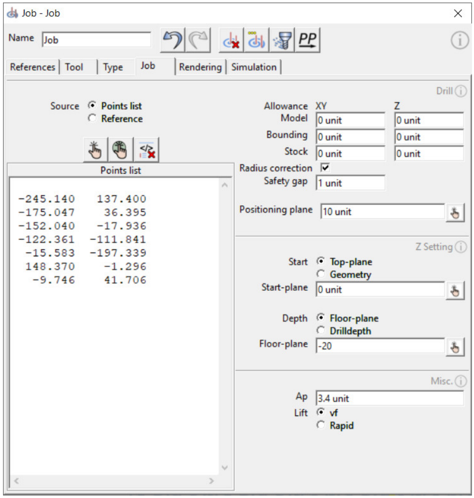
Parameter - drill

The "Drill" Job strategie is a semi-automatic cycle for calculating drilling movements.

Drilling positions can either be entered manually in a list or selected directly on the 3D model using various selection tools and transferred to the list. Alternatively, it is possible to transfer the drilling positions from another job.

> ***Note***
>
> As long as the job dialog is open, a preview of the drilling positions is displayed in the render window. This preview shows lines representing the drilling axes to give you a visual orientation of the planned drillings.

Precision
The values of the point list are transferred to the voxel grid during the calculation. As a result, the coordinates of the points in the output NC file may deviate from the coordinates entered in the point list. The accuracy of the drilling positions is determined by the project resolution.



Name: Job

Source: ● Points list ○ Reference

```
-245.140   137.400
-175.047    36.395
-152.040   -17.936
-122.361  -111.841
 -15.583  -197.339
 148.370    -1.296
  -9.746    41.706
```

Allowance	XY	Z
Model	0 unit	0 unit
Bounding	0 unit	0 unit
Stock	0 unit	0 unit

Radius correction ☑
Safety gap: 1 unit
Positioning plane: 10 unit

Start: ● Top-plane ○ Geometry
Start-plane: 0 unit
Depth: ● Floor-plane ○ Drilldepth
Floor-plane: -20

Ap: 3.4 unit
Lift: ● vf ○ Rapid

Source

Specifies the source from which the drill positions are obtained.

Points list

The points list contains the XY coordinates of the drilling positions, starting from the NC-Zero Point. Each line of the list defines a drilling position and consists of two values (position X and Y).

As soon as the input focus leaves the text field or the F5 key is pressed, the list is automatically formatted, cleaned up and checked for errors. The drilling axes displayed in the render window are also updated.

> ***Tip***
>
> Hole positions can also be created or generated in an external program such as a spreadsheet and then inserted into the point list via copy&paste. Separators and other special characters are removed by the automatic formatting.

Set drilling position
With this function, drill positions can be selected directly in the render window by mouse click. The selection mode can be terminated by right-clicking in the render window and calling up the context menu or by pressing the "Esc" key. The set points are added to the points list.

Set and center drill position
This function allows setting drill positions in the center of automatically detected shapes in the render window. The selection mode can be terminated by right-clicking in the render window and calling up the context menu or by pressing the "Esc" key.

To successfully select a shape, left-click the side face of the desired shape (e.g. the vertical inside of a hole). The shape must not intersect the model boundary.

Delete points list
This function removes all entered points in the text field.

Reference

The drill positions are taken from the selected reference.

If the drill positions of the reference change, they are automatically updated when the job is recalculated. Any job of the type "Drill" is allowed as a reference and must be located above the job in the project hierarchy.

Z Setting

In this section, the automatic detection of the drilling start points and the drilling depths are configured.

Start *[ppDrillTop]*
Defines the mode how the drilling start points are determined.

Mode	Comment
Start plane	All holes start on a plane. The specified value determines the height of the plane starting from the NC-Zero Point.
Geometrie	The starting point in direction Z of all holes is determined automatically. The surface of the geometry references, as well as the tool and the radius compensation are taken into account. An optional offset shifts the starting points by the specified value.

Depth *[ppDrillFloor]*
Defines the mode in which the drilling depths are determined.

Mode	Comment
Plane	All drill holes end on a plane. The specified value determines the height of the plane starting from the NC-Zero Point.
Drilling depth	All drill holes are the same depth. The depth results from the starting point and the specified value. A negative value must be specified.

Miscellaneous parameters

This section contains additional settings for drilling.

Vertical cutting depth *[ppApMain]*
After the tool reaches the specified vertical cutting depth, it performs a retraction movement to break the chip. It prevents chip jam by lifting out of the hole and repositioning itself in the hole. This process is repeated until the drilling depth is reached.

Lift
Defines the feed mode with which the tool lifts out of the hole after each cutting depth and repositions itself in the hole in the case of multiple infeeds.

> ***Tip***
>
> The vertical cutting depth is visually displayed at the calculated toolpath as a dashed line and can thus be easily checked.

Resulting stock

After successful calculation of the job, a resulting stock is added to the job as a subelement. This can be reused as a continuation blank for the next job.

The resulting stock is calculated from the referenced stock, the tool and the toolpaths. It is automatically deleted as soon as the job is recalculated or reset.

Collision report*[ppCollision]*

During the calculation of a job, a collision between the tool and the stock can be detected. In such a case, the system creates an object called "*_COLLISION" as a sub-element to the job. This object shows the contact points of the non-cutting tool geometry with the workpiece and is automatically inserted together with the resulting stock. If the job is recalculated or reset, the collision object is automatically deleted.

> ***Tip***
>
> In the tool editor, the mode for calculating the tolerance threshold can be changed.

For a collision check to be performed, the following conditions must be met:

- The collision check must be activated in the job.
- A tool must be referenced in the job.
- A stock must be referenced in the job.

A collision is detected when the tool makes contact with the stock during rapid traverse or when the non-cutting geometry of the tool makes contact with the stock.

Quick start with Lua scripting

Lua is a lightweight, efficient and easy-to-learn scripting language and is one of the most popular scripting languages for embedding in applications to provide users with an interface to customize and control the application according to their own needs.

The main advantage of Lua is its simplicity and flexibility. Lua code is easy to understand and requires minimal programming skills to get started. A Lua script does not need to be compiled before execution. This allows for quick and easy customization of the code as needed.

In VxCraft, Lua is used for various areas such as settings, tool scripts and especially postprocessors.

Basics

Even though Lua is a simple scripting language, a complete documentation would go beyond the scope of this manual. Therefore, this chapter focuses only on the absolute essentials and provides a brief overview of the basics of Lua. Nevertheless, it should be sufficient to make your own adjustments to existing scripts and to create simple scripts yourself.

For more in-depth information on Lua, there are numerous online resources that are easily accessible.

In the script examples in this chapter, the vxPrint() function is used to output to the screen. vxPrint() is a function provided by VxCraft that is used to display content in the postprocessor console. It is important to note that this function is not a native Lua function, but was developed specifically for use in VxCraft.

Comments

Comments are an important addition in a script to make the code more understandable and to leave hints and explanations to other developers or even to yourself. In Lua there are two types of comments:

Single line comments
Single-line comments start with two hyphens (--) and extend to the end of the line. Lua ignores everything after the hyphens and does not interpret it as code.

```
local myvar = 1+2 -- This is a one-line comment after an expression
```

Multiline comments
Multiline comments start with --[[and end with]]. Everything between these marks is treated as a comment and ignored by the interpreter.

```
local myvar_a = 1+2 --[[
This is a multiline comment
and extends over
several lines.]]
local myvar_b = 2+myvar_a
```

Lua case sensitivity

It is important to note that Lua, like many other programming languages, is case sensitive. This means that case is distinguished in variable names, function names, and keywords. For example, the variables myVariable and MyVariable are treated as two different variables. Therefore, always make sure to use the correct notation to avoid errors and unexpected behaviors.

Variables and data types

Lua supports various data types such as nil, number, string, boolean, table, function and userdata. Variables do not need to be declared and automatically get the data type when they are assigned a value. For example:

Data type	Example	Explanation
nil	local a = nil	An undefined value
number	local b = 42	A number
string	local c = "Hello, world!"	A character string
boolean	local d = true	A truth value (true or false)
table	local e = {1, 2, 3}	An array or associative array

Conditions and control structures

Conditions and control structures are fundamental elements of any programming language and allow you to control the program flow depending on conditions. In Lua there are several control structures, such as if-then-else, while and for loops.

if-then-else

The if-then-else structure in Lua is a control structure that allows you to execute different blocks of code based on one or more conditions. With this structure, you can control the flow of your program by providing alternative code paths depending on a particular condition.

The basic syntax of the if-then-else structure is as follows:

```
if Condition1 then
  -- Codeblock1 is executed if condition1 is true
elseif Condition2 then
  -- Codeblock2 is executed if condition1 is false and condition2 is true
else
  -- Codeblock3 will be executed if all previous conditions are false
end
```

The if-then-else structure begins with the if keyword followed by a condition that can be either true or false. If the condition is true, the code block is executed after the then keyword. If the condition is false, the code block is executed after the else keyword, if any.

The optional 'elseif' script block can be used to check additional conditions if the previous conditions are false. It is possible to use multiple elseif blocks in the if-then-else structure to check different conditions one after another. The structure ends with the end keyword.

The following table shows the comparison operators in Lua and their meaning:

Operator	Explanation
a <= b	is true if a is less than or equal to b
a < b	is true if a is less than b
a >= b	is true if a is greater than or equal to b
a > b	is true if a is greater than b
a == b	is true if a is equal to b
a ~= b	is true if a is not equal to b
a and b	is true if both a and b are true
a or b	is true if either a or b is true, or both are true

while loop

A while loop is used to execute a block of code as long as a certain condition is met.

Script example	Output

`local i = 1`
`while i <= 5 do`
`  vxPrint("Value from i: ", i, "\n")`
`  i = i + 1`
`end`
	`Value from i: 1`
	`Value from i: 2`
	`Value from i: 3`
	`Value from i: 4`
	`Value from i: 5`

for loop

The for loop is another type of loop that allows a block of code to be executed for a specified number of iterations. Lua supports numeric for loops and generic for loops.

Numeric for loop:

Script example	Output

`local i = 1`
`for i = 1, 5 do`
`  vxPrint("Value from i:", i)`
`end`
	`Value from i: 1`
	`Value from i: 2`
	`Value from i: 3`
	`Value from i: 4`
	`Value from i: 5`

Generic for loop:

Script example	Output

`local t = {"red", "green", "blue"}`
`for k, v in pairs(t) do`
`  vxPrint("Key: ", k, " Value: ", v, "\n")`
`end`
	`Key: 1 Value: red`
	`Key: 2 Value: green`
	`Key: 3 Value: blue`

Functions

Functions are reusable blocks of code that perform a specific task. They can accept parameters and return values. Functions are useful for better organizing code and avoiding repetitive tasks.

To define a function in Lua, use the keyword "function" followed by a name for the function and parentheses containing the parameters. The function body is terminated by the keyword "end".

Script example

function printPosition(x, y, z)
  vxPrint("Pos X: ", x, " Pos Y: ", y, " Pos Z: ", z, "\n")
end

To call the function with the desired coordinates, use the function name followed by parentheses and the corresponding parameters:

Script example	Output
printPosition(1, 3.5, 50.0)	Pos X: 1 Pos Y: 3.5 Pos Z: 50.0

The next example creates a function that rotates the X and Y coordinates of a position by a specified angle around the Z axis and returns the newly calculated coordinates.

Script example	Output

-- Rotates x and y by the specified
-- angle around the Z axis
function rotateZ(x, y, deg)
  local rad = math.rad(deg)
  local newX = x * math.cos(rad) - y * math.sin(rad)
  local newY = x * math.sin(rad) + y * math.cos(rad)
  return newX, newY
end	

local pos_x = 10 local pos_y = 20 local pos_z = 50 printPosition(pos_x, pos_y, pos_z) -- Rotates the coordinates by 45° -- using the rotateZ function: pos_x, pos_y = rotateZ(10, 20, 45.0) printPosition(pos_x, pos_y, pos_z)	Pos X: 10 Pos Y: 20 Pos Z: 50 Pos X: -7.071 Pos Y: 21.213 Pos Z: 50

Script flow and function calls

This subsection explains the basic flow of Lua scripts and how functions are used. It is important to understand how Lua scripts are executed and how functions are used in this flow.

A Lua script is executed from the top down. This means that the code is executed in the order in which it appears in the script. However, if a function is defined in the script, it will not be executed automatically. A function will only be executed if it is explicitly called in the script.

.

Note that functions must be defined **before** they are called in the script. Otherwise, Lua will not be able to find the function and an error will occur. So make sure your functions are always defined before their first call.

Here is a simple example to illustrate these concepts:

Script example	Output

`--Variablen`
`local home_x = -200;`
`local home_y =  200;`
`local home_z =  100;`
 	
`--Function which outputs a fixed defined home position`
`function printHome()`
`  vxPrint("Homeposition: ",`
`           home_x, " ",`
`           home_y, " ",`
`           home_z, "\n")`
`end`
 	
`-- Main part of the script`
`vxPrint("This is the beginning of the script.\n")`
`vxPrint("The following is the output of the home position:\n")`
`printHome()`
`vxPrint("after the output of the home position,\n"`
`        .."the script continues here.")`
 	
`-- End of the script`
	`This is the beginning of the script.`
	`The following is the output of the`
	`home position:`
	`Homeposition: -200 200 100`
	`after the output of the home position,`
	`the script continues here.`

In this example the code is executed from top to bottom. First, the variables home_x, home_y, and home_z are defined, then the printHome() function is defined to output the predefined home position. Then the main part of the script is executed, calling the printHome() function to output the home position. Finally, the last line of the script is executed, stating that the script continues here.

Settings module

config.ini und config.lua

The following initialization files are located in the "ini" subfolder of the installation path:

config.ini
This initialization file is called before all other settings files and determines the basic parameters.

- Specifies the path to the programdata folder (**see chapter Installation for more information about this folder**)
- Sets the default language selected in the setup

config.lua
This initialization file is called directly after config.ini and performs additional initializations.

settings.lua

By default, the settings.lua file is located in the **Programdata folder** and defines all basic and custom settings for VxCraft.

This file is a Lua script that is executed by VxCraft to set the preferences.

A description of the individual setting options is included in the script itself in the form of comments.

The module provides the following functions for setting the settings:

activate(string)

Function	activate(string)
Description	Activates the specified settings group. All subsequent settings are set in this group.
Example	activate("settings")

The function accepts the following values as arguments:

Argument	Description
"envar"	Activates the list of VxCraft environment variables for setting important paths.
"settings"	Activates the VxCraft settings. This group contains all other system settings.

set(string, string)
setoptional(string, string)

Function	set(string, string)
Description	Sets the specified setting in the active settings list. The first string is the setting key, and the second string contains the setting value. Multiple entries are overwritten, the last set setting is always valid.
Example	set("language", "en")

Function	setoptional(string, string)
Description	Works like set(), but the setting is only set if it does not already exist in the active list.
Example	setoptional("language", "de")

colors.lua

Works on the same principle as the settings.lua settings file. In this list, entries related to colors can be added or removed.

It should be noted, however, that this file is generated automatically by VxCraft. This means that all comments, user-defined formulas and functions are taken into account when the colors are read in, but are lost due to the automatic generation.

In other words, all formulas are converted into standardized entries.

Tool Script module<PRO>

This Lua module is used in the tool editor and provides functions to create the tool geometry. This module can be used to create construction elements such as lines and arcs, from which the rotation profile for the tool is generated. The orientation and dimensions of the construction elements are subject to certain restrictions:

- The elements must not cross the tool axis
- The elements must not create an undercut in both directions in the geometry

In other words, the end point of an element always has a larger value or the same value in one direction than the start point.

Script sequence

A tool script works according to the following principle: the script is processed from the top line to the last line. Before the command for the creation of an element is given, the parameters for this element are defined by functions. After creating an element, all parameters are reset and must be set again for the following element.

The tool is designed starting from the tool nose. The first element always starts at X0 and Z0. The geometry always ends at the maximum radius of the tool end, usually at the shank end.

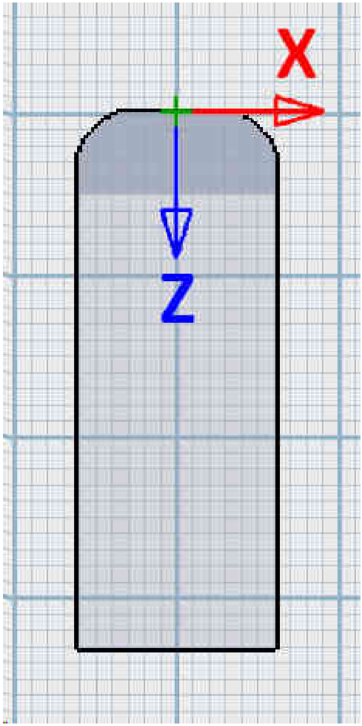
The following example script for the generation of a Bull End Mill will illustrate this:

```
to_x(3)        -- The next element
               -- ends at position X3
to_z(0)        -- The next element
               -- ends at position Z0
line()         -- Create a line with
               -- the set parameters

tangent()      -- The next one joins the
               -- last element tangentially
arc_rad(3)     -- The arc radius is 3
slope_out(0)   -- The exit angle is
               -- parallel to the tool axis
arc_a()        -- Create an arc in the
               -- clockwise

tangent()      -- The next one joins the
               -- last element tangentially
length(30)     -- and the line has a
               -- length of 30
line()         -- create the line with the
               -- set parameters

cutedge(5)     -- The cutting edge of the tool
               -- reaches up to Z5
```



© vxcraft.com

Bull End Mill

Parameter functions

All lengths are interpreted as user units.

Not all functions are compatible with each other, since some combinations can create contradictions. For example, it is not possible to specify an angle with slope_in() if the target point of a line has already been determined by to_x() and to_z() and the angle is derived from it.

Another example concerns an arc that is tangent to a line using the tangent() function. In this case, the angle of entry is already specified and cannot be additionally specified with slope_in().

Function	slope_in(grad)
Element	Arcs, lines
Description	Specifies the entry angle for the next element.
Value range	0°-90° 0°: The angle is parallel to the tool axis (vertical) 90°: The angle is horizontal
Example	slope_in(45) -- eg. for engraving tool tip

Function	slope_out(grad)
Element	Bögen
Description	Specifies the exit angle for the arc.
Value range	0°-90° 0°: The angle is parallel to the tool axis (vertical) 90°: The angle is horizontal
Example	slope_out(0) -- Element ends in vertical alignment

Function	tangent()
Element	Arcs, lines
Description	The next element joins the last element tangentially. This function determines the entry radius of the next element.
Example	tangent() -- Element connects tangentially.

Function	to_x(abs_position)
Element	Lines
Description	Specifies the absolute X target position of the line. The line does not necessarily have to run horizontally..
Value range	The target position must be greater than or equal to the start position.
Example	to_x(30) -- Line ends at position X30

Function	to_z(abs_position)
Element	Lines
Description	Specifies the absolute Z target position of the line. The line does not necessarily have to be vertical.
Value range	The target position must be greater than or equal to the start position.
Example	to_x(30) -- Line ends at position X30

Function	length(wert)
Element	Lines
Description	Determines the length of the line, regardless of the orientation
Value range	> 0
Example	length(5) -- The line has a length of 5

Function	arc_rad(radius)
Element	Arcs
Description	Determines the radius for an arc.
Value range	> 0
Example	arc_rad(3) -- The arc has a radius of 3

Generation

The following functions create the design elements based on the previously defined parameters.

Optionally, an error description string can be passed as an argument. If the element creation fails, this error description will be displayed as an error message. This facilitates the localization of errors in complex geometries.

Function	line(optional_error_string)
Element	Line
Description	Creates a line with the previously defined parameters. It is necessary to ensure that the value for the end point is greater than or equal to the start point in both directions.
Example	line("Shaft transition could not be created.")

Function	arc_a(optional_error_string)
Element	Clockwise arc
Description	Creates a clockwise arc with the previously defined parameters. Regardless of the calculation of the entry and exit angles, it must be ensured that the exit angle is **smaller** than the entry angle.
Example	arc_a("Arc for tool nose could not be created")

Function	arc_b(optional_error_string)
Element	Arc counterclockwise
Description	Creates an arc counterclockwise with the previously defined parameters. Regardless of the calculation of the entry and exit angles, it must be ensured that the exit angle is **greater** than the entry angle.
Example	arc_b("Arc for Roundover could not be created")

General functions

The following functions concern the tool in general.

Function	cutedge(position_z)
Description	The cutting edge of the tool extends to this height. The function is necessary for all tools. It is independent of other functions and can be located anywhere in the script.
Value range	> 0
Example	cutedge(12) -- The tool cutting geometry ranges up to Z12

Function	autocolltol()
Description	This command activates the automatic calculation of the collision tolerance. The collision tolerance is described in more detail in the chapter about **predefined tools**.
Example	autocolltol() -- Activates the automatic calculation of the collision tolerance

Postprocessor module

This module deals with the customization and creation of new postprocessors for VxCraft.

Postprocessor script

A postprocessor is essentially a Lua script file that contains special commands and functions to convert the toolpaths calculated by VxCraft into a suitable format for the CNC machine.

By default, the scripts are stored in the "PP" subfolder of the **(see chapter "Installation")** Program Data folder. New scripts can be added and managed in this folder.

Before you create or modify a postprocessor, please make sure that you save it under a unique name. Otherwise, when you update or install a new postprocessor package, your customized postprocessor could be overwritten. Please note that the prefix 'vxc_def_*' in the file name is used by VxCraft for predefined default postprocessors and therefore should not be used for your customized postprocessors.

Script sequence

The script flow in this module is different from other Lua modules because it is not linear.

The module is based on the concept of callback functions, which are provided by the user in the Lua script and called by VxCraft depending on the situation.

When a postprocessor run is executed, the script is first read and executed. Usually, only initialization tasks are performed in the script during this process. This run is necessary for VxCraft to capture the callback functions provided by the user.

Following this, VxCraft calls these callback functions as needed. For example, callJobBegin() is called at each job start and callMove(...) is used for each tool path point.

Functions

For the module, the following functions provided by VxCraft can be used and called anywhere in the script:

Function	vxPrint(...)
Description	vxPrint can be called with any number of arguments separated by commas. It converts all arguments to strings and prints them to the postprocessor environment console. The following data types are allowed: nil, number, string and boolean. **Note:** String literals such as %d, %s, %f, etc. are not considered by vxPrint().
Example	vxPrint("Start position: X ", pos_x, " Y ",4," Z ",50.0, "\n")

Function	vxClear()
Description	Deletes the contents of the postprocessor environment console.
Example	vxClear()

Function	vxWarn(string)
Description	Outputs the passed string as a popup warning message.
Example	vxWarn("The tool path is outside the maximum traverse path.")

Function	vxAsk(string)
Description	Outputs the passed string as a popup message that can be acknowledged with "Yes" or "No". Returns the value 1 if the message was acknowledged with "Yes".
Example	vxAsk("A collision was detected in the job. Continue?")

Function	getSetting(keystring)
Description	getSetting() returns the system setting with the passed key as a string.
Example	name = getSetting("pp_user")

Function	getSettingInt(keystring)
Description	getSettingInt() returns the system setting with the passed key as an integer.
Example	opendir = getSettingInt("pp_open_explorer")

Callback functions

The callback functions provided by the user must conform to a specific format. It is not necessary to provide all callback functions, and the order does not matter. If a function is missing, its absence is ignored and the process continues.

Return value
Each callback function must return "nil" if it runs without errors. If an error occurs, an error message can be returned in the form of a string. This aborts the postprocessor run and the message is displayed.

> ***Note:***
>
> In callback functions, as generally required in Lua, the correct upper and lower case must be observed.

Callback function prototypes

The listing below provides an overview and description of all available callback functions:

function callBegin()

Called at the beginning of the postprocessor run.If only a single NC file is to be created, this function can be used to prepare this file and open it in write mode.

function callJobBegin()

Called every time a new job starts.Can be used, for example, for changing the tool and for pre-positioning.

function callMove(x,y,z,gflag,mflag)

Called for each toolpath point.This is the only callback function that receives arguments from VxCraft.

x,y,z
The coordinates of the tool path point as a floating point number.

gflag, mflag
These arguments give additional info about the tool waypoint. They are flags in integer format that can be checked with the bit operator "&" for certain states using an if query. Global variables provided by VxCraft can be used for checking. In the following a listing with possible queries:

> ***Note:***
>
> The callMove() function is called very frequently and should therefore be limited to writing coordinates only. Computationally intensive tasks lead to a slow postprocessor run and an issued warning message by this function could be repeated countless times.

Check	Description
if(gflag&FLAG_G0)	The tool waypoint is part of a positioning movement. The regular rapid traverse feed should be used.
if(gflag&FLAG_G1)	The tool path point is part of a regular milling movement. The regular milling feed should be used.
if(gflag&FLAG_G1_RAMP)	The tool waypoint is part of a ramped plunge movement. The plunge feed should be used.
if(gflag&FLAG_G1_PLUNGE)	The tool path point is part of a vertical plunge movement. The drill feed should be used.
if(mflag&FLAG_M_SPINDLE_START)	The spindle should be started before the tool path point is approached.

Global variables

Global variables are provided by VxCraft and can be accessed anywhere in the script.All global variables are updated before callJobBegin() is called and are valid until after callJobEnd() ends:

Project and system related variables

global variable	Description
ppDirProject	Path to the saved project file
ppFilePath	Path to the postprocessor script
ppFileBase	Postprocessor script name
ppFileExt	File extension of the postprocessor script
ppNameProject	The name of the project element
ppNameNcPack	The name of the NC-Pack object
ppNameTool	The name of the tool
ppNameJob	The name of the job object
ppVxPerUnit	Project resolution in voxels per user unit
ppUnitPerVx	Project resolution in user unit per voxel

global flags	Description
FLAG_G0	Check value for gflag (see function "callMove()")
FLAG_G1	Check value for gflag (see function "callMove()")
FLAG_G1_RAMP	Check value for gflag (see function "callMove()")
FLAG_G1_PLUNGE	Check value for gflag (see function "callMove()")
FLAG_M_SPINDLE_START	Check value for mflag (see function "callMove()")

Job related variables

Start position:

global variable	Description
ppStartposX	Position X of the first waypoint of the current job
ppStartposY	Position Y of the first waypoint of the current job
ppStartposZ	Position Z of the first waypoint of the current job

Spindle and tool:

global variable	Description
ppSpindleRotation	The spindle direction set in the job, 0 = Clockwise, 1 = Counterclockwise
ppSpindlespeed	The spindle speed set in the job
ppToolId	NC tool number of the current job
ppCooling	Cooling mode of the current job

feed rate:

global variable	Description
ppFeedMode	The feed rate mode set in the job, 0=rapid, 1=user defined feed rate.
ppFeedRapid	The user-defined feed rate intended for the rapid.
ppFeed	Feed rate for regular milling.
ppFeedRamp	Feed rate for ramped plunge movements.
ppFeedDrill	Feed rate for vertical plunge movements.

Job parameters:

global variable	Description
ppJobtype	Calculation algorithm used
ppPlaneSite	positioning plane
ppPlaneStart	Start plane (top machining plane)
ppPlaneEnd	End plane (lowest working plane)
ppPlungeDrill	Vertical plunge active/inactive
ppPlungeRamp	Ramped plunge active/inactive
ppPlungeSpiral	Spiral plunge active/inactive
ppPlungeAngle	Angle for ramp and spiral plunge
ppPlungeDia	Spiral diameter
ppSgap	Safety distance
ppAreaIn	Zone Inside active/inactive
ppAreaOut	Zone outside active/inactive
ppAreaStk	Workpiece active/inactive zone
ppRadcorr	Radius correction active/inactive
ppAe	Horizontal cutting depth
ppApMain	Vertical main infeed
ppApMicro	Vertical cutting depth of the intermediate steps
ppDrillTop	Drilling mode top
ppDrillFloor	Drilling mode floor
ppAllm_xy	Allowance model XY
ppAllm_z	Allowance model Z
ppAllf_xy	Allowance bounding XY
ppAllf_z	Aufmaß Grenzmodell Z
ppAlls_xy	Allowance stock XY

ppAlls_z	Aufmaß Werkstück Z
ppFeedDir	climb=0 / conventional=1
ppOpt	Optimization mode
ppCollision	A value other than 0 means that a collision was detected in the job.
ppPause	Status of the pause option set in the job

Keyboard shortcuts and mouse control

Render view camera control

To change the viewing angle of the render view, one of the keys listed here must be pressed and held while moving the mouse cursor.

Key	Comment
left mouse button	Moves the camera linearly.
right mouse button	Rotates the camera.
Ctrl + right mouse button	Zooms in/out the camera section
Mouse wheel	Zooms in/out the camera section

General keyboard shortcuts

Key	Comment
Strg + X	Copies the selected objects to the clipboard and removes them.
Strg + C	Copies the selected objects to the clipboard.
Strg + D	Duplicates the selected objects.
Strg + V	Pastes the copied objects from the clipboard into the active project.
Strg + S	Saves the active project.
< | >	Toggles the visibility of the selected objects.

General keyboard shortcuts and mouse control for text fields

Key	Comment
F5	Checks the input and updates the render view if necessary
Enter	For single-line text fields the behavior is identical to F5
Arrow up	Increments the number of a parameter field with small spacing
Arrow down	Increments the number of a parameter field with small spacing
Page up	Increments the number of a parameter field with large spacing
Page down	Increments the number of a parameter field with large spacing
Mouse button left	With the left mouse button pressed, the value of a field can be incremented by moving the mouse up or down. This behavior can be deactivated in the settings.

File formats

VxCraft file formats

Format	Description
*.vxc	Project file
*.vxcfree	Project File Free Edition
*.vxct	Template file
*.lua	Settings file
*.lua	Postprocessor

Formats for importing and exporting 3D models:

Format	Description
*.stl	3D model in STL binary data format. The obsolete ASCII format is not supported.

Formats for importing and exporting 3D heightmaps:

Format	Description
*.png	Recommended. Due to the lossless compression of the PNG format, a clean 3D model can be created.
*.bmp	Provides lossless compression, but usually larger file sizes than PNG.
*.gif	Not recommended for detailed models due to 256 color limitation.
*.jpg	Not Recommended. The compression of the JPG format causes artifacts in the 3D model.
*.stl	export direct, import indirect possible

Third Party Components

VxCraft uses the following third-party components:

IUP, IM

https://www.tecgraf.puc-rio.br/iup/
https://www.tecgraf.puc-rio.br/im/
https://www.tecgraf.puc-rio.br/

Lizenz:
Copyright © 1994-2020 Tecgraf/PUC-Rio.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Lua

https://www.lua.org/

Lizenz:
Copyright © 1994–2023 Lua.org, PUC-Rio.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GLAD

https://glad.dav1d.de/

Lizenz:
Copyright (c) 2018, Salesforce
All rights reserved.

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

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

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

zlib

https://www.zlib.net/

Lizenz:
(C) 1995-2017 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

- The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
- Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
- This notice may not be removed or altered from any source distribution.

Jean-loup Gailly: jloup@gzip.org

Mark Adler: madler@alumni.caltech.edu

stb image loader

https://github.com/nothings/stb

Lizenz:
Copyright (c) 2017 Sean Barrett

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

IMPRINT

Published by Martin Kapfinger

E-Mail: info@vxcraft.com

Disclaimer: The publisher does not guarantee the accuracy, completeness or timeliness of the information in this manual and disclaims any responsibility for damages that may result from the use of the manual or the software described in it.

Vx
KRAFT