934 DT

Documentation

HdM Rhino Cutter Plug-In

Holger Rasch 25.11.2013

Plug-In description

It was the goal to create a new Plugin for the HdM Cutter and Laser cutter with the following requirements:

1. Shorter cutting times due job optimizations
2. Forecast of the needed cutting time
3. Directly implemented in Rhino without any external tool

Job Optimization

Without optimization of the path to cut, the cut order will be used in order of its creation. Especially for large jobs there could be a lot of empty tool traveling. To prevent this, we tested different algorithms to optimize the process order. We are using the so called Nearest-Neighborheuristic (NENE) followed by a 2-opt-heuristic. The NENE is a simple method of graph theory. Starting from the current position, the nearest neighbor is searched. The plug-in always considers the start point and the end point of an object. If necessary, the cutting direction is reversed (i.e., from end point to start point). The 2-opt heuristic is a de-crossing mechanism by turning the cut direction. Since both methods can be computationally very expensive, we applied - depending on the number of objects to be optimized - only the NENE or in the worst case no optimization, because an optimization calculation would take longer than the expected time savings. Can in principle lead to an NENE of any bad solution, together with the 2-opt-heuristic we never get a worse result than without optimization.

We tested also some other algorithm like the so called Farthest-Insertion and a simple Sort-by-X method. Both of them delivered only exceptional better results.

In the case that must comply the cutting direction with the drawn direction of the geometry you must omit the optimization.

Cutting Time

At three different situations a time statement will be shown:

Before Cutting: During the creating of the job preview the estimated time will be displayed in the form h:mm. These values based on a simulation of the length of the cut, empty travel paths and the known speed of the tools. The precision is limited to incalculable factors like calibrating of the device during the cut or while the tool is changing.

While cutting: Based at the already cut length, the already elapsed time and the still to cut length the time of completion will be shown (like 16:22). After every cut object, the result will be updated. The precision will increase to the end of a job. Empirically you can trust the result after a quarter of the job is done. Imprecision can happen if at the beginning of the job a completely different kind of geometries are cut as at the end (like a bunch of circle at the beginning and at the end a lot of boxes).

After Cutting: When the job is done, the duration of the job will be displayed.

Rhino Implementation

The plugin was deigned to work directly in Rhino with no other (external) tools. You will get a preview of the geometry which will be cut. You can set values for the cut job like cut speed etc. Finally the commands will be transmitted directly to the cutter. The cutter will be start without any further interaction. Inside Rhino you can follow the current job at the screen. The currently cut object will be highlighted at the screen and the remaining time will be displayed. It is still possible to prepare cut jobs, save them as plt-file and send them later to the cutter. These prepared file will be also shown on the screen after loading. A later change of the tool parameter is also possible.

All device depending parameters are defined outside of the plugin in some special configuration files. With these configuration files we are able to made later customizations, without any change on the program code.

APPENDIX A

for pin board next to the cutter and
as part of the handout documentation after the cutter introduction

Preparation

Rhino 5 File

Open your file with rhino 5. You can also import a .dwg with Rhino 5. The plug-in does not work with Rhino 4.

Scale/Units

Modify the scale. The drawing has to be drawn in millimeters at a model scale 1:1.

If you draw a 100 mm line, the machine is going to cut 100 mm.

Layers

Place what you want to cut to layers that you want. Please consider that you can assign each layer to only one tool. You can assign multiple layers to the same tool:

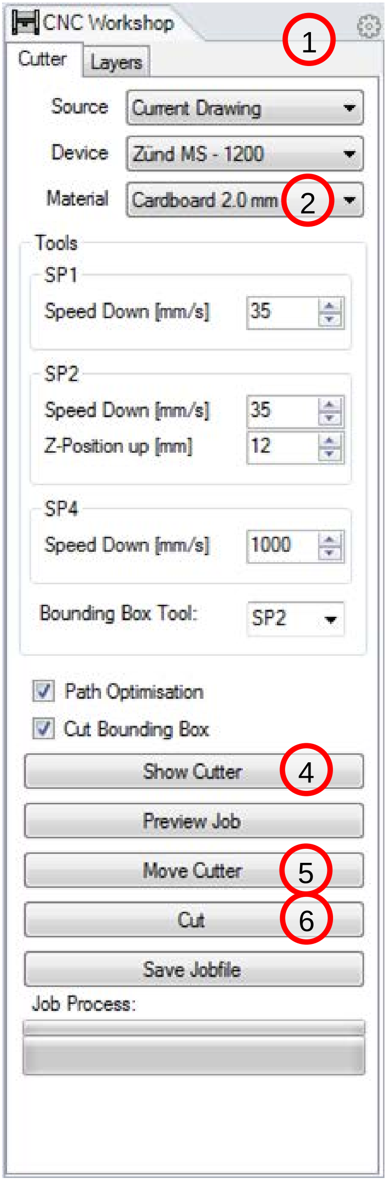
Layer 01 → SP1
Layer 02 → SP2
Layer 03 → -
Layer 04 → SP2
Layer 05 → SP4

There is no way to control the cut order of the tools. The plug-in internal optimization will sort the geometry in an own manner.

Start the Plug-In

If the Plug-In panel is not visible start the plug-In with the command *CNC_OpenPanel*. If Rhino regret this with a message like: “Unknown command: *CNC_OpenPanel*” please read the chapter “Plug-In Installation”.

How to cut from a drawing



Source: Current Drawing
Device: Zünd MS - 1200
Material: Cardboard 2.0 mm

SP1 Speed Down [mm/s]: 35

SP2 Speed Down [mm/s]: 35
SP2 Z-Position up [mm]: 12

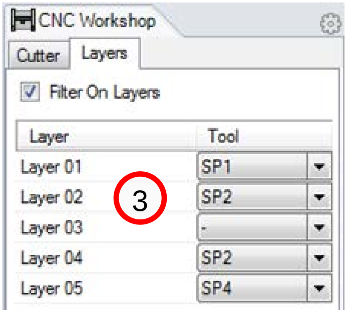
SP4 Speed Down [mm/s]: 1000

Bounding Box Tool: SP2

☑ Path Optimisation
☑ Cut Bounding Box

1. Make sure that you can see the whole plugin with all buttons like in the picture left.

2. Select a material to get the best settings for the tools



☑ Filter On Layers

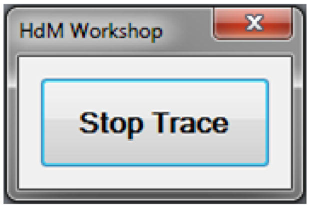
Layer	Tool
Layer 01	SP1
Layer 02	SP2
Layer 03	-
Layer 04	SP2
Layer 05	SP4

3. On the Layer Tab make the right assignment which layer will be cut with which tool.

4. Show the Cutter; so that you can:

5. Move it to your geometry.

6. Cut!



While you cut, you will see in Rhino which geometry is currently cut, how long it takes etc. As long as you are tracing the job progress, you can’t work with Rhino. You can stop the trace. This will **not stop** the cut or the job transmission to the cutter.
You cannot stop the cutter in Rhino.

APPENDIX B

as part of the handout documentation after the cutter introduction
and even hanging as bound print-out next to the cutter

Work with the Plug-In

The plugin contains two tabs. One with all settings related to the cutter device, and one with the layer tool assignment.

Cutter Tab:

1. With the “Source” list you can select from where your geometries come that you want to cut. It can be the current drawing or a previously prepared plt-file. In the last case an “Open file” dialog appears to select a plt-file.

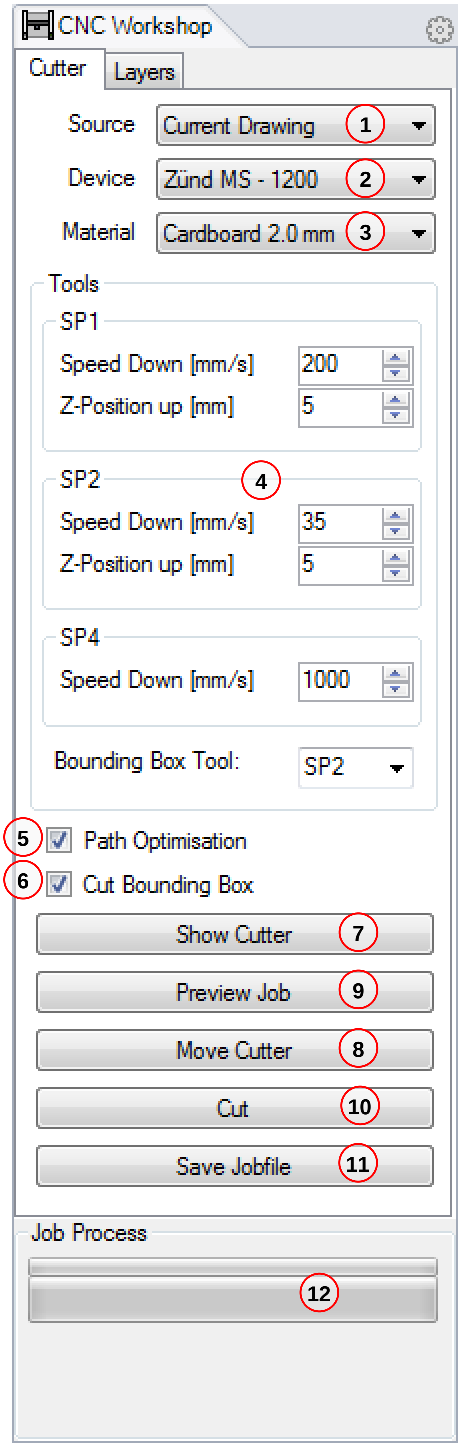
2. If you cut from the current drawing, you have to decide, on which cutter you want to cut. You can choose between “Zünd MS-1200 Cutter” and the “Eurolaser 800”. Once you loaded a plt file, you can’t change the device anymore.

3. From the Material list you can select some predefined material settings. These values are well approved, but you can change this of course.

4. In the “Tools” area you can change several tool settings like cutting speed, travel height and also the tool to cut a bounding box.

5. With the path optimisation you can save a lot of cutting time. The plugin calculate a short path through your geometry without unnecessary empty traveling paths. Unfortunately this procedure is very computationally intensive. With more than 1500 objects the calculation process will take longer than the possible time saving. In this case the path optimization will be switched of automatically. Because the optimization algorithm can turn the cut direction, you have the opportunity to switch it off.

6. By default a bounding box with an offset of 5 mm will be cut around your job. This is to save material because a well cut rest of material is easier to recycle. If your job is very big and the rest of your sheet is really useless you can switch off the bounding box.



Source: Current Drawing
Device: Zünd MS - 1200
Material: Cardboard 2.0 mm

Tools

SP1
Speed Down [mm/s]: 200
Z-Position up [mm]: 5

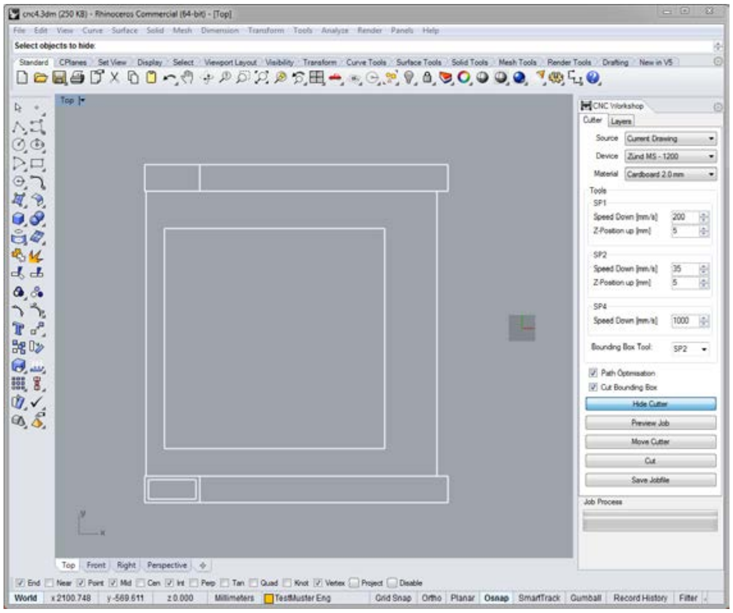
SP2
Speed Down [mm/s]: 35
Z-Position up [mm]: 5

SP4
Speed Down [mm/s]: 1000

Bounding Box Tool: SP2

☑ Path Optimisation
☑ Cut Bounding Box

7. Show Cutter – this will display the device in Rhino. At the first time the cutter will be shown at 0,0,0 .



cnc4.3dm (250 KB) - Rhinoceros Commercial (64-bit) - [Top]

Select objects to hide:

CNC Workshop

Source: Current Drawing

Device: Zünd MS - 1200

Material: Cardboard 2.0 mm

Tools

SP1
Speed Down [mm/s]: 200
Z-Position up [mm]: 5

SP2
Speed Down [mm/s]: 35
Z-Position up [mm]: 5

SP4
Speed Down [mm/s]: 1000

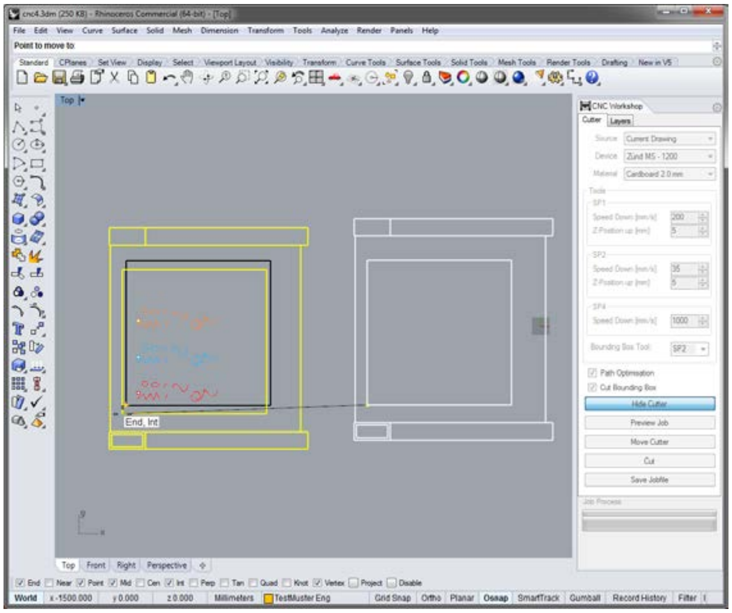
Bounding Box Tool: SP2

☑ Path Optimisation
☑ Cut Bounding Box

World x 2100.748 y -569.611 z 0.000 Millimeters TestMuster Eng

File Edit View Curve Surface Solid Mesh Dimension Transform Tools Analyze Render Panels Help
Cutter
Layers
Hide Cutter
Preview Job
Move Cutter
Cut
Save Jobfile
Job Process
Top
Front
Right
Perspective
Grid Snap Ortho Planar Osnap SmartTrack Gumball Record History Filter

8. If your geometry to cut is not located at 0,0 you can move the geometry to the device, or you move the device to the geometry with the “Move Cutter” button. A left click will start a move operation like in a standard rhino move operation. You have to choose from which point to which point do you want to move the cutter. You can only snap points at the cutable area. A right mouse click starts a move operation from the left bottom corner to a point. If you have loaded a plt-file, you can move the position of your job instead of the device.

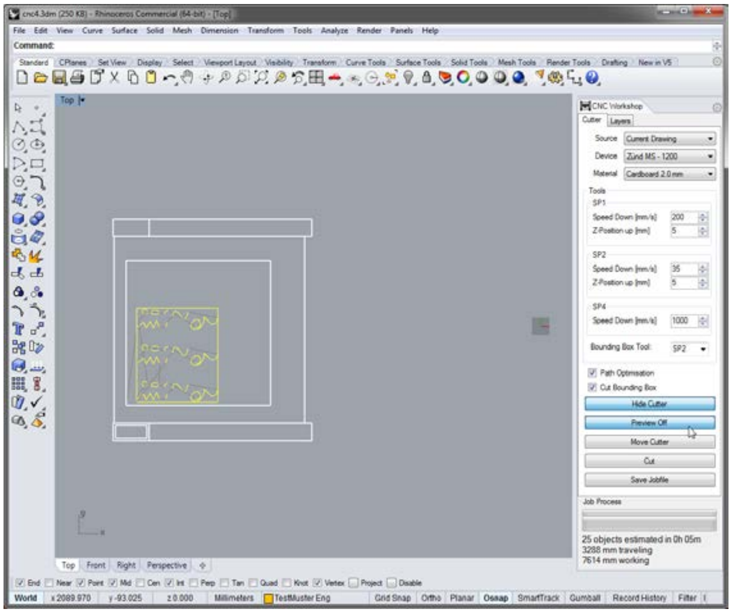


cnc4.3dm (250 KB) - Rhinoceros Commercial (64-bit) - [Top]

Point to move to:

World x -1500.000 y 0.000 z 0.000 Millimeters TestMuster Eng

9. With the “Preview Job” button you will see in yellow which objects will be cut. Gray lines indicate the non-cutting traveling of the tool. If nothing will be displayed, please check in the layer tab the right setting (see 16.). During the preview is on, all objects are invisible except the cutter geometry. In the bottom line of the plugin you can see the estimated time for the job.



cnc4.3dm (250 KB) - Rhinoceros Commercial (64-bit) - [Top]

Command:

Source: Current Drawing

Device: Zünd MS - 1200

Material: Cardboard 2.0 mm

SP1
Speed Down [mm/s]: 200
Z-Position up [mm]: 5

SP2
Speed Down [mm/s]: 35
Z-Position up [mm]: 5

SP4
Speed Down [mm/s]: 1000

Bounding Box Tool: SP2

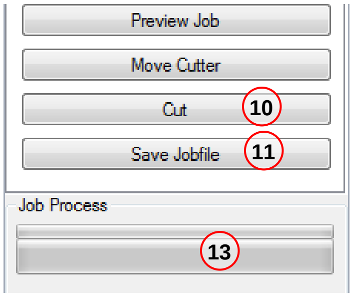
☑ Path Optimisation
☑ Cut Bounding Box

Job Process

25 objects estimated in 0h 05m
3288 mm traveling
7614 mm working

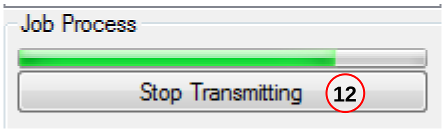
World x 2089.970 y -93.025 z 0.000 Millimeters TestMuster Eng

10. If everything looks fine you can send the job direct to the device by clicking “Cut”. If you click with the left mouse button you will trace the job progress on the Rhino screen. A right mouse button click will send the job command directly without tracing.

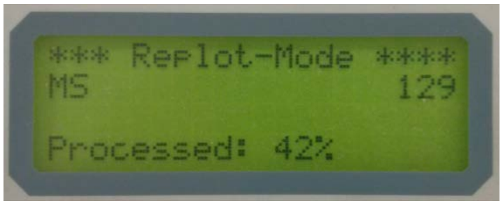


11. “Save Jobfile” will write a plt-file for later use. A save file dialog will appear.

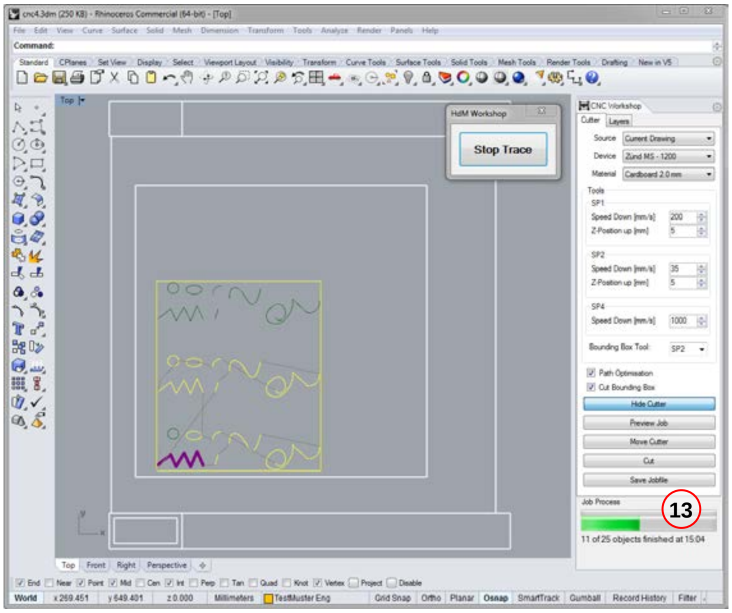
12. While the job command is sent to the cutter you can stop the transmission.



13. During the cutting process you can see which object is currently cut (in purple) which object will be cut later (yellow) and which objects are already cut (green). In the bottom line of the plugin you can see the estimated finish time. A progress-bar shows the current job progress and in the fourth line of cutter display you see the percent that has already been cut.



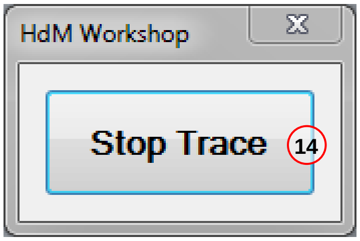
```
*** Replot-Mode ****
MS              129

Processed: 42%
```



Source: Current Drawing
Device: Zünd MS - 1200
Material: Cardboard 2.0 mm
Tools
SP1
Speed Down [mm/s] 200
Z-Position up [mm] 5
SP2
Speed Down [mm/s] 35
Z-Position up [mm] 5
SP4
Speed Down [mm/s] 1000
Bounding Box Tool: SP2
☑ Path Optimisation
☑ Cut Bounding Box

11 of 25 objects finished at 15:04

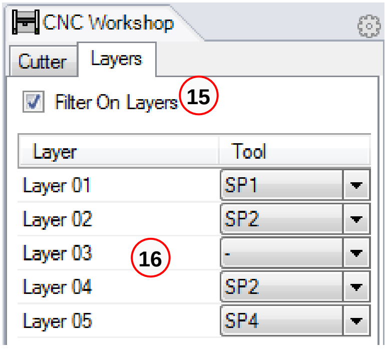
14. If you do not want to follow the cutting progress in Rhino click “Stop Trace”. While tracing you cannot use Rhino. This will not stop the cutting process or the job transmission to the cutter. **You cannot stop the cutting machine in rhino!**



Layers Tab

15. You can filter all visible layers

16. On the Layers tab you have to assign which layer is using which tool. This information will be stored in the drawing.



☑ Filter On Layers

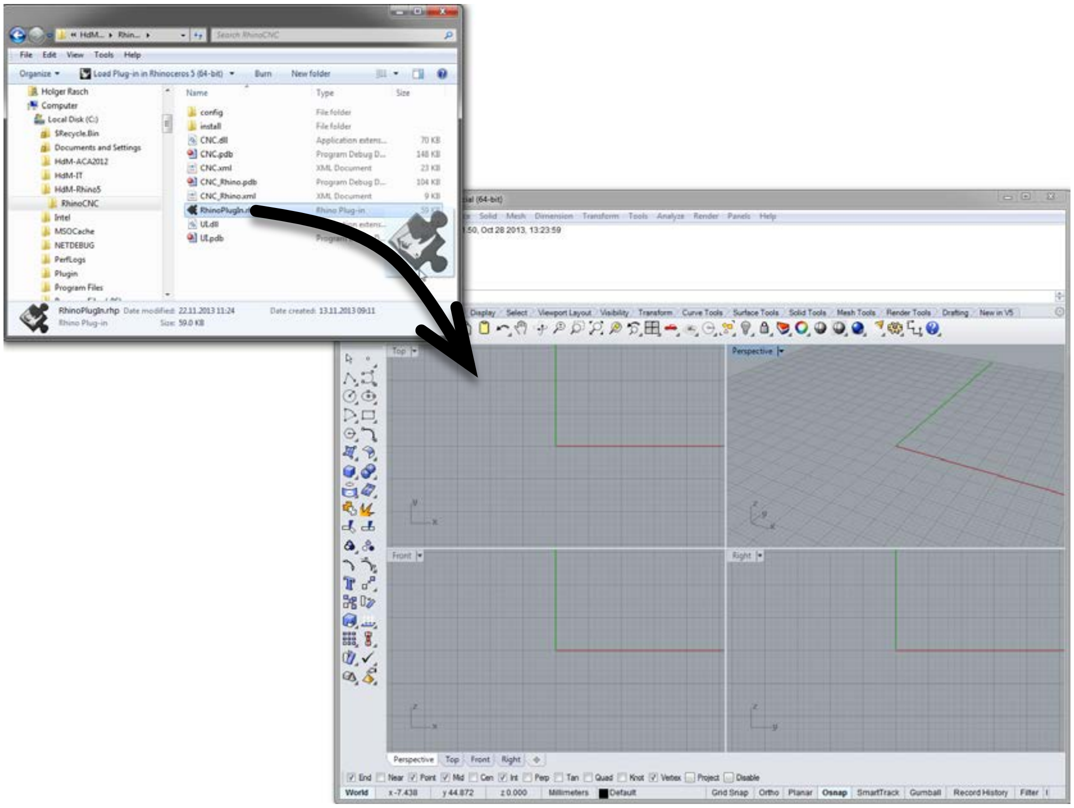
Layer	Tool
Layer 01	SP1
Layer 02	SP2
Layer 03	-
Layer 04	SP2
Layer 05	SP4

Plug-In Installation

The plugin is already installed at the control computers next to the cutters.

If you want to work on your own computer or something is failed on the control computer try the following steps:

1. Close Rhino
2. Make sure that nobody else is logged into the computer.
   If you're not sure please restart the computer.
3. Open Rhino5
4. Drag & Drop the file "C:\HdM-Rhino5\RhinoCNC\RhinoPlugIn.rhp" to the Rhino5 window



Name	Type	Size
config	File folder	
install	File folder	
CNC.dll	Application extens...	70 KB
CNC.pdb	Program Debug D...	148 KB
CNC.xml	XML Document	23 KB
CNC_Rhino.pdb	Program Debug D...	104 KB
CNC_Rhino.xml	XML Document	9 KB
RhinoPlugIn.r	Rhino Plug-in	59 KB
UI.dll	...tion extens...	
UI.pdb	Program	

RhinoPlugIn.rhp Date modified: 22.11.2013 11:24 Date created: 13.11.2013 09:11
Rhino Plug-in Size: 59.0 KB

5. In Rhino5 type the command: CNC_OpenPanel
6. Make sure that you can see the whole plugin with all buttons.

If you cannot find the folder C:\HdM-Rhino5\RhinoCNC do the following:
Open a Windows Explorer an go to: "W:\01 AUTOCAD\ACA2012\admin”.
If you can’t see this folder, type the whole path in to the address bar.
Double click on “Sync force - HdM-Rhino5.bat” and try this instruction again from point 1.

APPENDIX C

For DT and if necessary IT use only

Preparation for the control computer

A control computer is a computer with the ability so send cut commands directly via serial COM port to a cutter. Without the following steps the plugin cannot send job to the device, only plt file creation is possibly.

The following instruction is only for the control computers which are directly connected to the cutter devices!

Every control computer has to be defined in the hdm.workshp.xml file. Read the chapter about the workshop xml. Important is the controller entity:

```
<controller pair="M1200@BSL-DE35-1062" connnection="Serialport 1"/>
```

M1200 is an ID from a device.xml and BSL-DE1062 the computer name of the control computer. “Serialport 1” is the ID for the connection also defined in the workshop.xml file.

You can find your computer name by pressing ⊞ + Pause

The following both operation must execute with local administration rights:

In “C:\HdM-Rhino5\RhinoCNC\install” you find a shortcut to start rhino together with the `CNC_OpenPanel` command. You must replace the two local shortcuts under: “C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Rhinoceros 5” and “C:\Users\Public\Desktop”.

Additionally is it necessary to import the “C:\HdM-Rhino5\RhinoCNC\install\Plugin_Install.reg” to the registry. These registry settings will be register the plugin so that he install it for every user automatically and will also call `CNC_OpenPanel` if a user double clicks on a rhino file.

XML Documentation

All parameters for the Plug-In are defined in a bunch of XML-files. They are located in the config folder at the same location as the RhinoPlugIn.rhp file. There are three different kinds of files. The filename is delimited in three parts. At first a simple description followed by a point, one of the three types (workshop, device and material) and at last the file extension xml. For a proper work we need at least one of each type.

After finalizing our plugin had these files:

hdm.workshop.xml	–	main description
zund.device.xml	–	description for the Zünd cutter
laser.device.xml	–	description for the Laser cutter
cardboard.material.xml	–	description for the cardboard material
foamboard.material.xml	–	description for the foamboard material
plexi.material.xml	–	description for the plexi material

*.workshop.xml

The workshop-file contains all global information about our “workshop”. This is currently a list of all control-computers and how they are connected to the computer. And also stored here are the definitions for the communication setting. Additionally the command to switch a cutter online is also sored here. It can only exist one .workshop.xml file, like: hdm.workshop.xml

Example for a minimal workshop.xml file:

```
<?xml version="1.0" encoding="utf-8" ?>
<workshop DefaultDevice="M1200">

       <connector type="serial" id="Serialport 1" port="COM1" speed="19200" parity="0"
       databits="8" stopbits="1" handshake="2" readtimeout="500" writetimeout="2000"/>

       <controller pair="M1200@BSL-DE35-1062" connnection="Serialport 1"/>

</workshop>
```

Root entity: `workshop` single mandatory attribute: `DefaultDevice`

The `DefaultDevice` is an ID defined from a device.xml file and descripts the device which will be used, if no precisely statements will follow.

Example: `<workshop DefaultDevice="M1200">`

Sub entities: `connector, controller`

The connector entity descripts a kind of connection. Currently only serial connections are supported. The following attributes are mandatory:

`type`	currently only “serial” is allowed
`id`	a unique ID for this entity
`port`	the Portname of the used communication port, like COM1
`speed`	serial baud rate
`parity`	parity 0 = none 1 = odd 2 = even 3 = mark 4 = space
`databits`	length of data bits per byte
`stopbits`	0=none 1 = one 2 = two 3 = OnePointFive
`handshake`	0=none 1 = XOnXOff 2 = RequestToSend 3 = RequestToSendXOnXOff
`readtimeout`	timeout in millisecond
`writetimeout`	timeout in millisecond

Example: `<connector type="serial" id="Serialport 1" port="COM1" speed="19200" parity="0" databits="8" stopbits="1" handshake="2" readtimeout="500" writetimeout="2000"/>`

The `controller` entity descripts a computer how’s controlling the cutter device. The following attributes are necessary:

`pair`	a pair indicates which computer belongs to which device in the form DeviceID@Computername
`connection`	an ID of a valid `connector` (see above)

Example: `<controller pair="M1200@BSL-DE35-1062" connnection="Serialport 1"/>`

*.device.xml

The device-file is the core description file. For each cutter we need one file. The file stores information about size, graphical representation inside Rhino and HPGL command phrases. Every tool of a cutter has here his description.

It can exist as much device.xml files as needed. Currently at HdM we have two of them. Per file is only one `device` entity allowed.

Example for a minimal device-file with just one tool:

```
<?xml version="1.0" encoding="utf-8" ?>
<device id="EX2000" text="Example Device 2000">

   <property id="factor" value="100"/>
   <property id="workzone" x="2000" y="2000"/>
   <property id="BoundingTool" text="TOOL"/>
   <property id="BBoxOffset" value="5"/>
   <property id="TresholdNonOptimization" value="10000"/>
   <property id="TresholdNearestNeighbourOptimization" value="1200"/>
   <property id="ToolOrder" text="TOOL"/>
   <property id="prejob" text="PU;PA;PB2,1;SD300;AS2,4;RS0,0,0;"/>
   <property id="postjob" text="JB0;XX12,2;MSDone by {username};NR;"/>
   <property id="SwitchOnline" text="ZF5;"/>

   <conduit id="absolute">
      <line x1="0" y1="0" x2="1200" y2="0"/>
      <line x1="2000" y1="0" x2="1200" y2="2000"/>
      <line x1="2000" y1="2000" x2="0" y2="2000"/>
      <line x1="0" y1="2000" x2="0" y2="0"/>
   </conduit>

   <tool id="TOOL">
      <property id="speeddown" value="20" min="0.1" max="100" text="Speed Down [mm/s]"
      displayfactor="10" ishidden="false"/>
      <property id="pretool" text="XX12,2;MSTool:{id};{id};ZP2000,0;VS{speeddown},100;CR2.0;"/>
      <property id="posttool" text="PU;"/>
   </tool>

</device>
```

Root entity: `device` mandatory attributes: `id, text`

`id` it’s a unique name for the device, this id will be used several times in the xml-file and the plug-in too to identify this device

`text` this is a free text which will be used in the graphic user interface instead of the id because it is more readable

Example: `<device id="M1200" text="Zünd MS - 1200">`

Allowed sub entities: `property, conduit, tool`

The `property` entity means a common property. In the device.xml it will used for a lot of different things:

`id`	the ID
`value`	a Value (Double)
`x`	a X coordinate value (Double)
`y`	a Y coordinate value (Double)
`text`	a text (String)
`min`	a minimal allowed value for this property. If the `value` attribute is smaller than `min`, the value will set to `min` (Double).
`max`	a maximum allowed value for this property. If the `value` attribute is bigger than `max`, the value will set to `max` (Double).
`ishidden`	decide that a `value` will be shown in the GUI (and will be changeable). This works only in sub entities of the `tool` element (Boolean).
`displayfactor`	the factor for the `value` to show in the GUI (Double)

Example: `<property id="speeddown" value="20" min="0.1" max="100" text="Speed Down mm/s]" displayfactor="10" ishidden="false"/>`

In a device.xml file, the following `id`s will be allowed inside of the `property` entities:

`factor`	between mm und den device units (Cutter) (value)
`workzone`	the size of the cutting area in mm (x, y)
`BoundingTool`	the ID of the default tool to cut a bounding box, it can be overridden by a material definition. (text)
`BBoxoffset`	offset of the bounding box from the geometry in mm. (value)
`TresholdNonOptimization`	value from which number of objects no path optimization will performed (integer)
`TresholdNearestNeighbourOptimization`	value from which number of objects only the nearest neighbor heuristic will performed (integer)
`ToolOrder`	comma separated list of tool ids in order to cut (text)
`delay`	time in millisecond that will be wait before the job will start, this is only effective if the `prejob` is using this, it will also use by the calculation for the time forecast (integer)
`prejob`	a chain of HPGL command which will send at the beginning of a job (text)
`postjob`	a chain of HPGL command which will send at the end of a job (text) **For a proper work is it very important that the `postjob` contains the `JB0;` and the `NR;` command.**
`SwitchOnline`	command to switch the device online if its send as front-end command

IMPORTANT: `prejob, postjob, pretool` and `posttool` has the capability to exchange values in curly brackets with values defined outside if the property. Actually this property was designed to send a HPGL command phrase before or after geometry based sequences. But these phrases must contain values controlled and defined by the user via the plugin interface. Therefore we can define properties which will be read by the plugin and after that they can be manipulate within the plugin. These manipulated values will be used to generate the output phrase. Value in curly brackets will be interpreted as id from a property and will replaced with his value.

Example:

`<property id="prejob" text="PU;PA;SD{delay};RS0,0,0;"/>` will be produced with the 300 from the `<property id="delay" value="300"/>`. this: `"PU;PA;SD300;RS0,0,0;"`

The placeholder {username} is not defined in an xml-file, he will replace with the current username.

Example:

`<property id="postjob" text="JB0;XX12,2;MSDone by {username};NR;"/> ;"/>` will be produced with the current logged in username, this: `"JB0;XX12,2;MSDone by h.rasch;NR;"`

The `conduit` entity descripts the graphical representation for the device inside of the Rhino workspace. Is has just one attribute (`id`) and even just one kind of sub entity:

`line` it’s a definition of a line from `X1, Y1` to `X2, Y2`. unit is millimeter (double)

Sub entity: `tool`, mandatory attribute `id` descripts a tool

The following properties have these ids:

`speedup` speed for the lifted tool in cm per second (double)

`speeddown` speed for the lowered tool in cm per second (double)

for Zünd only:

`zposup` Up-position for the tool in device units (double) (see `factor`) not for SP4

`zposdown` Down-position for the tool in device units (double) (see `factor`) not for SP4

for laser only

`powerwork` laser power while cut (percent, double)

`powermin` laser power while pen up (percent, double)

`powerrecess` laser power during the 'tool down after waiting time' (percent, double)

`interuptiontime` time between two cuts, this value is needed by time calculation (milliseconds, integer)

`pretool` a chain of HPGL command which will send at the beginning of a tool. **For a proper work is it very important that the `pretool` contains the `XX12,2;MSTool:{id};` command string.**

`posttool` a chain of HPGL command which will send at the end of a job.

Example:

```
…
<property id="delay" value="300" />
<property id="downacceleration" value="2" />
<property id="upacceleration" value="4" />
…
<property id="prejob"
text="PU;PA;PB2,1;SD{delay};AS{downacceleration},{upacceleration};RS0,0,0;"/>
…
```

The final output will be:

```
PU;PA;PB2,1;SD300;AS2,4;RS0,0,0;
```

*.material.xml

In a material-file we define names and tool settings for different materials. Every material that we define needs a unique ID. All settings with the same ID will be collected as same material. It is only necessary to define tool settings which differ from settings defined in the device files. It is not necessary to differ in the material definition between different devices. Only tool related settings are possible.

It can exist as much material.xml files as needed. It’s a good idea to create for each kind of material a separate file, but this is not mandatory.

Example for a simple material:

```
<materials>
   <material id="foamboard_3mm" display="Foamboard 3 mm">
      <toolsettings id="TOOL" speeddown="50"/>
   </material>
</materials>
```

Root entity: `materials`, no attributes

Sub entity `material` mandatory attributes: `id, display` optional attribute: `BoundBoxTool`

`id` unique id of the material, it has to be unique for all material.xml files
`display` name of the material how it will displayed in the Plug-In
`BoundBoxTool` id of the tool (see *.device.xml) to cut the bounding box.

Sub entity `toolsettings` mandatory attribute: `id`

`id` ID of the tool which we override a value

The following attributes are allowed. The values will override the definition in the device.xml

`speedup, speeddown, zposup, zposdown, pretool, posttool, powerwork, powermin, powerrecess, interuptiontime`