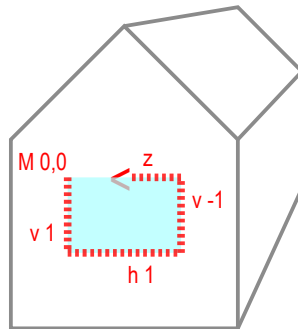


A mini-guide for Sweet Home 3D users

Editing SVG and other advanced furniture properties

Version 1.2
© Ola-Kristian HOFF, 2013
[CC-BY](#)



Contents

Purpose.....	2
Models (furniture) and file types.....	2
SVG.....	3
SVG and advanced properties used in SH3D.....	3
How to add SVG to SH3D models.....	3
SVG co-ordinates used in SH3D.....	4
SVG path commands.....	4
Examples.....	5
Full rectangle (5)	
..door with no threshold (doorframe with three sides) (5)	
Part rectangle cut-out (6)	
L shape(d staircase) (6)	
Door with diagonally slashed top (under angled ceiling) (6)	
An arch door (7)	
Arc cut-out 90°: pie-slice (7)	
Double cut-out arc 90° (7)	
Double cut-out arc 180° (7)	
A staircase that curves at the top (8)	
Spiral staircase (8)	
Keyhole cut-out (8)	
Visualising your SVG path in a browser.....	8
Sample file - text to copy/paste to <xxxx>.svg (9)	
More editing: PluginFurnitureCatalog properties.....	9
Example window entry with opening indicated (and outside ornaments)	
Example lamp entry – light source	
Information, legal and copyright notice.....	11
References.....	12
SVG references and tutorials	
SH3D links for SVG and catalogue properties	
Terminology and colours	
Legal links	

Purpose

This document is intended for [Sweet Home 3D](#) (SH3D) users who import models (furniture) and need to modify advanced properties with the *Furniture Library Editor*. To use this manual, you should use your own library of models and edit them with the FurnitureLibraryEditor available on the [SH3D download page](#).

The guide attempts to describe how to edit advanced properties for furniture, especially the SVG cut-out properties in the FurnitureLibraryEditor (or in the .properties file). Using basic SVG paths will allow you to customise how SH3D cuts holes in walls and ceilings/floors. This is useful if you want to use a model staircase, window or door that does not cut the wall, ceiling or floor exactly how you want it to. The document also briefly describes [editing the library .properties](#) file if you need to access the full range of properties. For most users this may not be very interesting. However, if you need very accurate staircases and windows/doors, you may need to look beyond the excellent, normal SH3D functions.

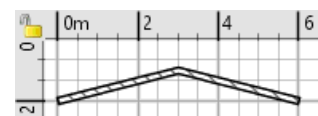
Thanks to the excellent functions of SH3D, most models are correctly managed without the need for advanced model editing. But IF you need customised models with a high level of accuracy, this guide could be of help understanding what you can do with your models.

Models (furniture) and file types

Sweet Home 3D and the FurnitureLibraryEditor let you import models from [.obj](#), [.dae](#), [.3ds](#) and [.lws](#) files. Models in these formats are widely available. The easy way to use these models is simply to download the model (often as a zip file) and import in SH3D (by dragging and dropping the file where you want it).

Note that there may be copyright restrictions on the models, if you put a copyrighted model in your home, you may be in violation if you make your home available to others. While the direct import of models with SH3D is sufficient for normal use, the FurnitureLibraryEditor lets you create your own collections and manage them with more advanced properties. It is also a neat way of managing the furniture you use frequently. Some points worth noting when working with models:

- You can create/edit models with a range of **3d modelling software**, such as [Art of Illusion](#), [Blender](#) and [Sketchup Make](#) (all currently free), and many others. Bear in mind that drawing in three dimensions on a two dimensional screen can be challenging. Use the tutorials and videos. Make sure you export your models to [.obj](#) ([Wavefront .obj file](#)) or [.dae](#) ([Collada](#), an XML schema) format or another of the supported formats: [.3ds](#) / [.lws](#).
- You can even **create basic models with SH3D** e.g. by drawing walls, exporting them to [.obj](#) and re-importing to SH3D adjusting rotation and size. This is especially useful for simple models (like roofs): draw a wall (angle = roof angle), export as [.obj](#), import and *rotate*/resize – and you have your roof. For more elaborate roofs, see the [slopingCeilingGuide.pdf](#).
- An **.obj (object) file** should be accompanied by an **.mtl** (material) file that defines colours, opacity and possibly image files with the texture of the various materials (or you can use the [material names recognised by SH3D](#)). The [.obj](#) (object) file links to an [.mtl](#) (material) file for the materials, the [.mtl](#) file links to the image files (such as [.png](#), [.jpg](#), [gif](#)) for textures. [.obj](#) / [.mtl](#) and image files can all be zipped into one file (which facilitates importing and handling).
- Beware of **file sizes**. Larger imports could impair SH3D performance. Rule of the thumb: the more complex the model, the larger the file size. For textures SH3D recommends maximum 512*512 pixels (but [256*256](#) or less is usually more than enough for a good result).
- Images (textures) can be edited** with freely available software, such as the image editors that come with every operating system or advanced image editors like the [GIMP](#) (also free and multi-platform).
- The 3d formats you can import with SH3D, however, do not contain all necessary information to exploit them fully in SH3D. For this you *may* need to access the more advanced properties.
- The furniture library, keeps a **list of properties on how SH3D shall use the different models**, dimensions rotation, wall and floor/ceiling cut-out etc. This list is stored in a file called [PluginFurnitureCatalog.properties](#).



The main model properties can be edited from SH3D. In the FurnitureLibraryEditor you can change all the properties available from SH3D, but also cut-out for windows/doors and staircases. To do this, some understanding of SVG is necessary.

SVG

Scalable Vector Graphics (SVG) is a [W3C](http://www.w3.org) format for 2d drawings. SVG can be created with a text editor or with software that creates SVG files based on a drawing (like [Inkscape](http://inkscape.org)), see en.wikipedia.org/wiki/SVG. The SVG elements can be included in web documents (HTML). That makes SVG a powerful tool for integrating graphics in web documents in a very compact format. Below only the PATH element will be described. SVG path commands are used to "scissor" customised holes in walls and ceilings/floors in SH3D.

SVG and advanced properties used in SH3D

For the purposes of SH3D, suffice to say that SVG is a way of describing the geometry needed to cut-out staircases (in ceiling and floor above) or windows/doors (in walls).

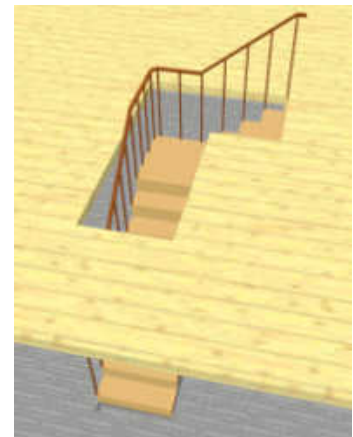
- When furniture is specified as *door/window*, SH3D will automatically:
 - *either* make the hole based on SVG from the Furniture Library file (written with other properties in a file called PluginFurnitureCatalog.properties),
 - *or* calculate the correct cut-out equal to the model shape (from SH3D 4.2).
- When furniture is specified as *staircase* in SH3D, the software will cut a hole in the ceiling / floor above equal to the the rectangle surrounding the staircase.

For most models of doors, windows and staircases, this works just fine. Insert a round window, and the wall will fit snugly around it, insert a rectangular staircase, and there will be a rectangular opening in the ceiling and floor above, etc.

However, sometimes you may need a hole that is different from the model. For instance for an angled or curved staircase – or a door with no threshold (threshold is bottom part of the door: the door-sill) to take the most obvious examples.

For the hole to be cut, the staircase must reach the ceiling (adjust height). The door/window must be aligned to the wall (SH3D aligns automatically when door/window is dragged and dropped on a wall and magnetism is active).

If you need the window/door to protrude from the wall (have frames) or contain lines showing how it opens (sash axis and angle), you need to use a door or window from the catalogue. *Or* you can edit PluginFurnitureCatalog.properties manually and add the required properties, how to do this is briefly described under [More editing](#).

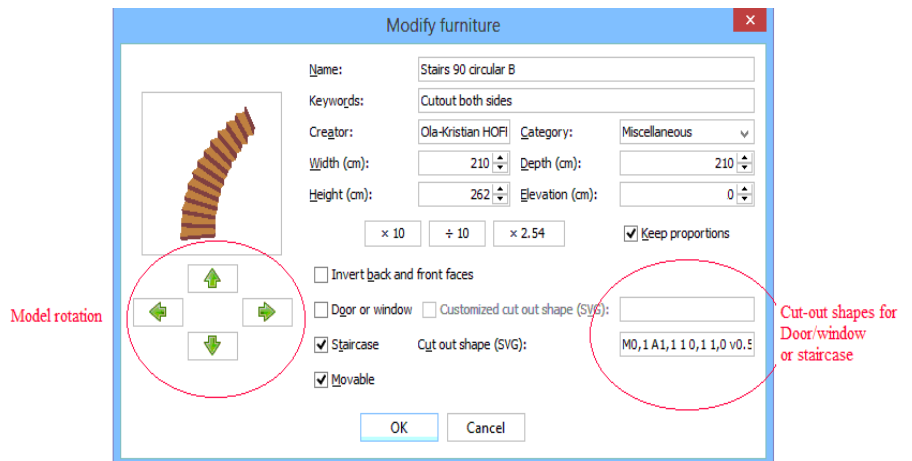


Example: staircase L-shape cut-out, without SVG specified, a large rectangle would be cut-out.

How to add SVG to SH3D models

To add your own cut-out shapes you need to use FurnitureLibraryEditor from the [SH3D download page](#), import the model(s) into the Library, edit the properties, save the library and import it into SH3D.

The FurnitureLibraryEditor allows you to edit many of the properties that SH3D uses for the furniture (but not all, see [More editing](#)). If you edit your models (library) with the FurnitureEditor you get a more elaborate window than within SH3D. Most notably the SVG cut-out for stairs and holes in the wall...

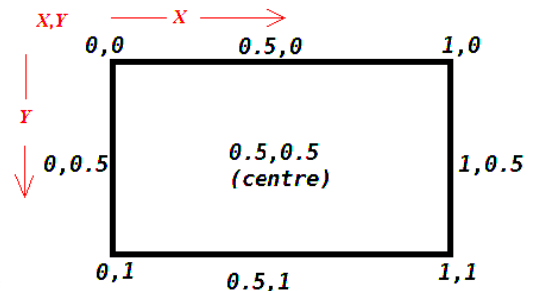


The following sections tries to describe how to enter the SVG paths for doors, windows and staircases. If your SVG path syntax is incorrect, the entry turns red (but it will not crash anything if you get it wrong, even if the cut-out does not work). Below you just find the very basic path commands. Visit the [W3C pages on SVG](#) for a more complete (and more correct) explanation. If you feel confused, that is normal, under [References](#) below you will find alternative links to take your confusion to a higher level.

Do not be disheartened by co-ordinates and commands, the [examples](#) are easier to understand.

SVG co-ordinates used in SH3D

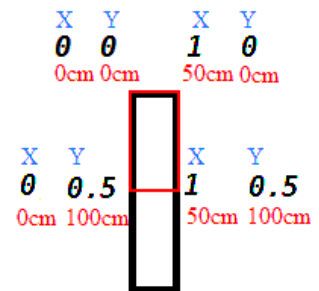
Imagine your model surrounded by a rectangle. Think of this rectangle as a co-ordinate system, a grid if you like. x,y tells you where you are in this rectangle. The upper left corner is 0,0 (0 on the X axis and 0 on the Y axis), the lower right corner is 1,1 (1 to the right, 1 down). The first figure (x) indicates horizontal position, the second figure (y) indicates vertical position.



Note the decimal values: comma separates the two co-ordinates, full stop is used as a decimal separator (decimal mark). Very un-European, but there you go. Also, SH3D will not let you cut outside the rectangle defined by the width and height of your model (workaround: add invisible lines to your model). And never mind the proportions of the illustration rectangle. For windows/doors, the rectangle is vertical: width (x) and height (y), for staircases horizontal: width (x) and depth (y).

SVG co-ordinate values used in SH3D go from 0 to 1 (the SVG viewBox). The 0 to 1 interval is, in other words, a relative value that corresponds to the real dimension. Suppose your model is 50cm wide and 200 cm high. You want to cut a rectangle in the top half, all the way (from 0 cm to 50 cm) horizontally and half way (100 cm) vertically. That means:

- all the way across (horizontally) = from 0 to 1 on the X axis (in this case 50 cm)
- half way down (vertically) = from 0 to 0.5 on the Y axis (in this case 100 cm)
- etc.



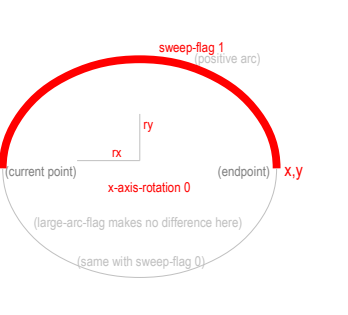
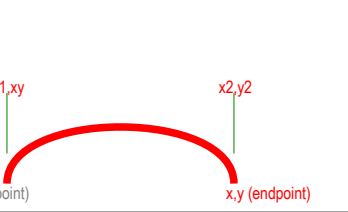
Entire model:
50cm wide * 200cm high

SVG path commands

SVG lets you draw (cut) "paths" within the shape of the model by moving from point to point – either in straight lines or curves. Typically, the default cut-out will look like this: `M0,0 v1 h1 v-1 z`. Think of it as a treasure map with instructions telling you how many steps to walk in each direction:

`M0,0 v1 h1 v-1 z` just means: start from top left corner (`M0,0`), go down one (`v1`), go right one (`h1`), go up one (`v-1`), return to start (`z`). See, not exactly black magic. This command will simply *cut around the entire rectangle*. But of course, it can be made more complicated by using more elaborate path commands (red means numbers that follow each command, and *this is not a complete list*). Hold your breath and just ignore what you do not understand below. It is easier to understand from the [examples](#) that follow.

M x,y (or m x,y)	Moveto: moves the "scissor" without cutting to co-ordinates x,y or relative: move x horizontally and y vertically without "cutting", upper-case (Absolute) / lower-case (relative) explained at the end of the table
L x,y (or l x,y)	Lineto: draws (=cuts) from current point to x,y (or horizontal x , vertical y).
H x (or h x)	Horizontal lineto: draws (=cuts) horizontally from current point.
V y (or v y)	Vertical lineto: draws (=cuts) vertically from current point.
	Note: strictly you do not need the H and V commands as you can do the same and more with the L command. The L command also lets you cut straight lines in <i>every</i> direction, not just horizontal/vertical lines. However, H and V are used in SH3D default shapes and are convenient for their simplicity.

<p>A <i>rx,ry x-axis-rotation large-arc-flag,sweep-flag, x,y</i> (or <i>a....</i>)</p>	<p>Elliptical Arc starts where "scissor" is (current point) and is defined by</p> <ul style="list-style-type: none"> - two radii (<i>rx,ry</i>) - <i>x-axis-rotation</i>, - <i>large/small,sweep</i> (0 or 1 values) (<i>0,1</i>= small, positive arc, <i>0,0</i>= small, negative arc etc.) - <i>x,y</i> (endpoint) <p>(the centre is calculated from above values)</p>	
<p>C <i>x1,y1 x2,y2 x,y</i> (or <i>c....</i>)</p>	<p>Curveto: Draws a curve (cubic Bézier) from current point to <i>x,y</i> with two control points which pull the curve in place. The control points <i>x1,y1</i> and <i>x2,y2</i> are placed outside the curve like magnets to pull the curve into shape.</p>	
<p>z</p>	<p>Closes the path (draws a straight line back to starting point).</p>	
<p><i>Upper-case / lower-case commands</i></p>	<p>Indicates whether the co-ordinates following the command are ABSOLUTE co-ordinate values (referring to the entire co-ordinate system) or relative to the current position: positive values go <i>right</i> or <i>down</i>, negative values go <i>left</i> or <i>up</i>. For instance: L 0.1,0: draw line to co-ordinates <i>x0.1 / y0</i> (from current point). l 0.1,0: draw line <i>0.1</i> right (from current point, on the same level: <i>0 y</i>).</p>	

So the approach when making a cut-out in SH3D is:

1. The commands will start with **M***x,y* to set the starting point of your cutting exercise. If you want to start your cut-out in the upper left corner, start with **M0,0**. If you want to start half way down the left side, start with **M0,0.5** etc.
2. Use commands to start drawing/cutting (specify lines from point to point, e.g. **v1 h1 v-1**)
3. And you end with a **z** – which draws (cuts) a straight line back to where you started (**M***x,y*).

Examples

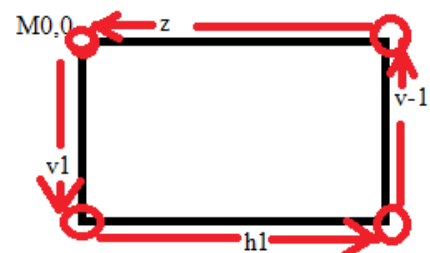
In the following examples, BLACK shows the rectangle surrounding the object, RED indicates the cut-out. You can cut either clockwise or anti clockwise.

Full rectangle

This example cuts the entire area of the object (default shape).


M0,0 v1 h1 v-1 z means:

M0,0	Move to co-ordinates 0,0
v1	Cut 1 vertically (down)
h1	Cut 1 horizontally (right)
v-1	Cut 1 vertically (- means up)
z	Cut to start (close)



This shape is identical to the shape of the rectangle surrounding the model. But it is still useful, for instance for a...

..door with no threshold (doorframe with three sides)

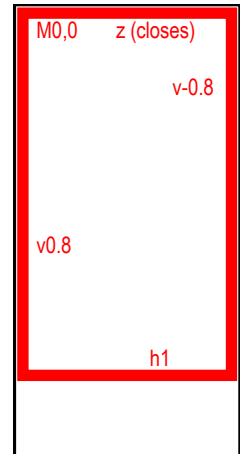
From version 4.2 SH2d will cut an opening equal to the shape of a door/window. So if the model is not closed, e.g. if the model only has three sides, like this , the cut-out will follow the shape and not display an opening in the wall. If you want the opening to surround the rectangle surrounding the model, you must add an SVG path for a rectangle: **M0,0 v1 h1 v-1 z**. This cut-out shape, in other words, forces a rectangle cut-out around the model.

Part rectangle cut-out

This example also cuts a rectangle, but only the upper 80% of the object.

M0,0 v0.8 h1 v-0.8 z (example is anti clockwise)

M0,0 Moveto co-ordinates 0,0
v0.8 Cut 0.8 vertically
h1 Cut 1 horizontally
v-0.8 Cut -0.8 vertically (- means up)
z Cut to start (closes)



This shape is useful for instance if you have a rectangular staircase where you want the hole in the ceiling/floor above to be covering only part of the staircase. For instance, you have a 300cm high ceiling, and you really do not need to cut a hole as long as you have 200cm headroom in the stairs. Assuming the staircase starts at the bottom of the model (and is a 45° angle), you need to cut roughly 2/3 down, 1 across, 2/3 up again and close: **M0,0 v0.67 h1 v-0.67 z**

Note 1: 200cm headroom is an example of a legal requirement for a staircase. But with 200cm of headroom instead of 300cm, you still get an extra square metre or so extra floorspace on the level above.

Note 2: Also, you may have realised that for accurate values, you may need to use the actual measures of your model and multiply with the co-ordinate value to find out the exact cm cut-out. For instance, if the staircase above is 400 cm deep, a cut-out of 0.8 will cut $400\text{cm} * 0.8 = 320\text{cm}$. The cut-out is saved with the model in SH3D in relative values, so if you later change your staircase to 360cm depth, the cut-out will scale to $360\text{cm} * 0.8 = 288\text{cm}$ (and that could mess with legal requirements for stairs angle etc.).

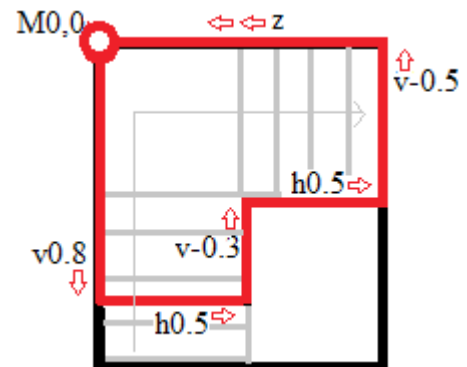
L shape(d staircase)

This example command cuts down 80% of the shape, across 50%, up 30%, across 50%, up 50% - and closes back to origin:

M0,0 v0.8 h0.5 v-0.3 h0.5 v-0.5 z (example goes anti clockwise)

means

M0,0 moves to origin 0,0 (top left corner of model)
v0.8 - down (vertical) 0.8
h0.5 - right(horizontal) 0.5
v-0.3 - up (vertical, negative) 0,3
h0.5 - right (horizontal) 0.5
v-0.5 - up (vertical, negative) 0.5
z - closes (line back to 0,0)



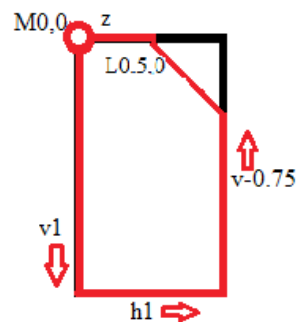
This shape was to cut out a staircase (grey) starting at the bottom left

(leaving some floorspace above the first steps by not cutting all) and then following the rest of the shape of the Γ formed staircase for the opening. SH3D would cut the entire black rectangle around the object if a cut-out is not added.

Door with diagonally slashed top (under angled ceiling)

M0,0 v1 h1 v-0.75 L0.5,0 z - cuts off the top right corner of a rectangle:

M0,0 - moves to origin 0,0 (top left corner of model)
v1 - down 1 (100%)
h1 - right 1 (100%)
v-0.75 - up 0.75 (75%)
L0.5,0 - draws a line to (absolute) co-ordinates 0.5x 0y
z - closes (line back to 0,0)



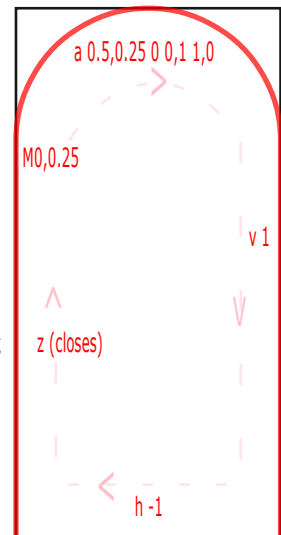
In old Scandinavian houses, you come across these doors under loft ceilings. Note: SH3D will cut this door automatically *if* the model is closed on all sides (has a threshold).

An arch door

`M0,0.25 a0.5, 0.25 0 0,1 1, 0 v0.75 h-1 z` – rectangle with arc on top (*clockwise*)

`M0,0.25` - moves to 0,0.25 (25% down) to start
`a0.5, 0.25 0 0,1 1, 0` - draws an arc 1,0 (1 across 0 down), 0.5/0.25 radii, sweep up
`v0.75` - draws a line down 0.75 (the remaining 75% down)
`h-1` - draws a line -1 (100% left)
`z` - closes with line back to M0,0.25

If this door model is closed at the bottom, SH3D will do a perfect job automatically. However, if the model has no threshold (no bottom sill, like this: \cap), SH3D will cut along the frame, thus closing the opening. To avoid this, add this path in the FurnitureLibraryEditor (adjust values depending on proportions). This example moves down 25%, then cuts *clockwise* around.

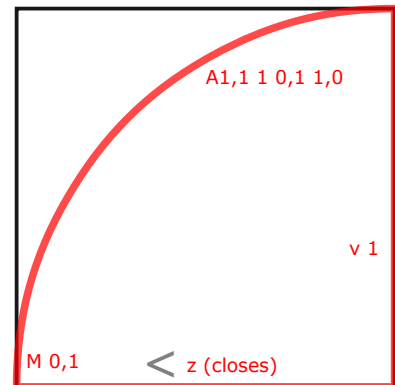


Arc cut-out 90°: pie-slice

As seen above, arcs obviously have more parameters than a straight line, this example shows a quarter circle (pie-slice), an arc from 9 o'clock to 12 o'clock (270° to 0°, or W to N for normal navigators). While the Arc is indeed more challenging, let us look at a simple example where you basically do not have to worry about the middle values (x-axis-rotation, large-arc-flag and sweep) in the Arc command. Arc starts at 0,1 lower left corner, radii are both 1 and endpoint is 1,0 upper right corner:

`M0,1 A1,1 1 0,1 1,0 v1 z` means

`M0,1` Moves to lower left corner (where the arc will start)
`A1,1` (the two arc radii: 1,1 – both equal to side of the rectangle)
`0` x-axis rotation (0° rotation)
`0,1` Large arc flag 0 and Sweep flag 1 (0,0 would *sweep* the arc the other way).
`1,0` ARC ENDPOINT (top right, 1 on x axis, 0 on Y axis)
`v1` Draws line from where arc ended to 1,1, lower right corner)
`z` closes (back to start-point M0,1)

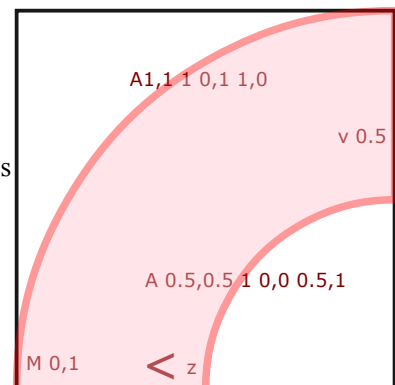


This cut-out could be used for a staircase turning in a quarter circle (but leaves the inner part of the stairwell open, see the double arches below if you want to close on both sides).

Double cut-out arc 90°

`M0,1 A1,1 1 0,1 1,0 v0.5 A0.5,0.5 1 0,0 0.5,1 z`

Cuts the same arc as above, but slashes only half way down the right side, draws a second arc back to the centre of the bottom line, and closes. See illustration below. Used to cut both sides of a quarter circle staircase.

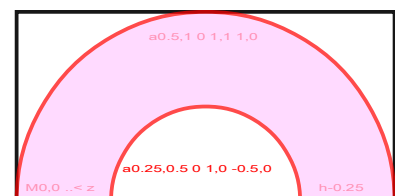


Double cut-out arc180°

`M0,1 a0.5,1 0 1,1 1,0 h-0.25 a0.25,0.5 0 1,0 -0.5,0 z`

Double arc as above, but with ½ circle and not just ¼.

Start bottom left, arc to bottom right, go 25% left, arc 50% left, close.





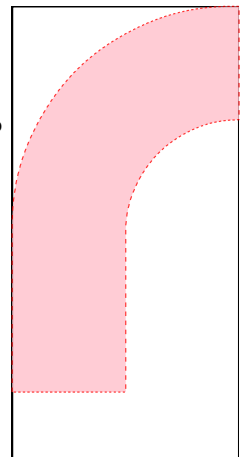
Door with slashed top
 Staircase arc with no cut-out
 Double 90° arc cut-out
 Round window
 Pie-slice cut-out 90°
 Double 180° arc cut-out
 Window w/edges
 Window w/ ornaments

A staircase that curves at the top

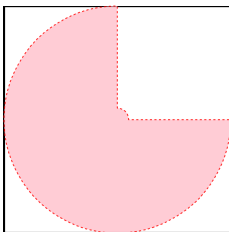
M 0,0.85 1 0,-0.35 A 1,0.5 1 0,1 1,0 10,0.25 A 0.5,0.25 0 0,0 0.5,0.5 1 0,0.35 z

Start 85% down, line up 35%, Arc to top right, line down 25%, Arc to centre, line down 35% and close.

Bottom 15% of the staircase will not be covered by the cut-out.



Spiral staircase



M 1,0.5 A 0.5,0.5 0 1,1 0.5,0 v 0.45 A 0.05,0.05 0 0,1 0.55,0.5 z

Start half way down the right side (**M 1,0.5**), cut an elliptical arc with radii half model size to middle of top (**A 0.5,0.5 0 1,1 0.5,0**), cut down 45% (**v 0.45**), draw a *small* arc with radii 0.05 and negative sweep-flag draw around the middle column (**A 0.05,0.05 0 0,1 0.55,0.5**) and close. This leaves ¼ of the staircase covered by ceiling/floor.

Keyhole cut-out

M 0.4,0.85 1 0.05,-0.5 a 0.12,0.12 1 1,1 0.1,0.0 1 0.05,0.5 z

Cuts a keyhole shape in the middle of the model. The practical use for this – doubtless far fetched – would be if you have a model of a keyhole (to be placed in a wall) with a round plate on one side and nothing on the other side. The automatic cut-out would follow the round shape on the bigger side of the object, leaving an empty hole shaped like a circle on the other side. To avoid that, you would need to cut out only the keyhole.



Visualising your SVG path in a browser

Trying (and failing) is inevitable. Going through the whole procedure to test changes in your cut-out is cumbersome. Here is how you can visualise in a (SVG enabled) browser.

1. Take the text below and paste into a plain text document (plain text, *not* formatted).
2. Save it as `<whatever>.SVG`.

3. Drag and drop the file into your browser.
 - Edit the path commands and save.
 - Refresh the browser.
 - Repeat until you are happy with the shape.
4. Then copy the commands (on one line) into the SVG field of FurnitureLibraryEditor.

Sample file - text to copy/paste to <xxxxx>.svg

```

<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="6cm" height="6cm" viewBox="0 0 1 1" xmlns="http://www.w3.org/2000/svg" version="1.1">
  <title>SH3D test SVG</title>
  <desc>OKH: SVG browser visualisation</desc>
  <!-- ANOTHER KIND OF SVG COMMAND, TO DRAW THE RECTANGLE AROUND YOUR PATH -->
  <rect x="0" y="0" width="1" height="1" fill="none" stroke="black" stroke-width="0.01" />

  <!-- REPLACE EXAMPLE AND FILL IN YOUR OWN PATH COMMANDS BELOW -->
  <path d="

      M 0.05,0.10
      L 0.05,0.30
      a 0.55,0.25 -15 1,1 0,0.5
      V 0.95
      h 0.90
      v -0.85
      z

      "
      fill="none" stroke="red" stroke-width="0.02" />
</svg>

```



RESULT IN BROWSER

The example assumes a square model. If you need a different rectangular shape, you can use a different SVG definition, for instance:

```

<svg width="5cm" height="10cm" preserveAspectRatio="none" viewBox="0 0 1 1"
  xmlns="http://www.w3.org/2000/svg" version="1.1">

```

You *can* also open the .svg file with Inkscape or another SVG editor. In fact, you can open it with other image software (raster) too, like the GIMP, but then you cannot save the file back to SVG. Even if software like Inkscape writes SVG, it may not be of much help as the co-ordinate system is bound to be different and the paths are on the complicated side to be of use for the relatively simple SH3D cut-outs. You can also make a 2d drawing in SH3D (for instance a room) and export it to SVG, but the co-ordinate system will be different than what is needed to make models. There are also online SVG editors (e.g. svgeditor.7jigen.net), but again, you probably need to work with a different co-ordinate system.

More editing: *PluginFurnitureCatalog.properties*

The FurnitureLibraryEditor is an excellent tool. However, there are still more properties that can be assigned to your models. For instance, the default catalogue contains lamps with lights, windows with openings indicated, half open doors etc. Also, for the neoclassical window (right bottom illustration above), you would need to do some extra editing to ensure that the ornaments actually protrude from the wall. These properties cannot be edited with the FurnitureLibraryEditor.

However, if you wish to be courageous, you can edit them manually following the general rough outline below. If you do not understand what this is all about, you probably should not be messing with this.

1. Import (drag-drop, obj/dae file) model into the FurnitureLibraryEditor. Save and back-up your library (*.sh3f file). I recommend using a copy library and import it into SH3D after editing.
2. The *.sh3f file is a zipped file with a directory for each model and thumbnails and a file with model properties – PluginFurnitureCatalog.properties – in the root directory of the zipped file.
3. PluginFurnitureCatalog.properties is a text file where each model is assigned several mandatory and optional [properties](#). The [DefaultFurnitureCatalog.properties](#) file uses the same format and is an excellent cheat-sheet. Pay close attention to the numbering and punctuation. My most common mistake is unintentionally using comma (and not period/full stop) as a decimal separator. Bad idea.

4. Add and make changes, add a lightsource, window properties (both light and window properties cannot be put into the same model :-() etc. [This forum thread](#) explains the window and door properties (useful for putting frames around the doors and windows and adding ornaments).
5. Put PluginFurnitureCatalog.properties back in the sh3f file and re-import the library into SH3D. With a bit of luck, it actually works.

You probably want to be careful when you edit the properties and feel comfortable with editing code files. They can be quite unforgiving, a wrong comma could render the file useless (although, FurnitureLibraryEditor and SH3D will normally just disregard a wrong entry or wrong properties).

Example window entry with opening indicated (and outside ornaments)

```
id#1=okh#Window_urban_old
name#1=Window Urban Old
tags#1=Neoclassical ornament window
category#1=Doors and windows
icon#1=/Window_urban_old.png
model#1=/Window_urban_old/window_urban_old.obj
multiPartModel#1=true
width#1=200
depth#1=60
height#1=220
movable#1=true
doorOrWindow#1=true
doorOrWindowCutOutShape#1=M0.24,0.22 v0.71 h0.52 v-0.71 z (cuts inside window ornaments)
doorOrWindowWallThickness#1=40
doorOrWindowWallDistance#1=2
doorOrWindowSashXAxis#1=50
doorOrWindowSashYAxis#1=28
doorOrWindowSashWidth#1=100
doorOrWindowSashStartAngle#1=0
doorOrWindowSashEndAngle#1=-90
elevation#1=30
creator#1=Ola-Kristian HOFF
```

MODEL ID: author#model_id
NAME (as used in SH3D)
TAGS: searchable key words
CATEGORY (as used in SH3D)
ICON: the mini-image in SH3D
(link to model: .obj file inside library)

|
| MODEL DIMENSIONS
|

|
| Shows hinges and opening
| Multiple values possible
|

SH3D will calculate the depth of the window (normally equal to wall thickness), you will need to set additional parameters if the window frame should protrude from the wall (on one or both sides). That is determined based on the relationship between Depth#, doorOrWindowWallThickness# and doorOrWindowWallDistance#. You can add several numbers on the sash properties for "double" windows. See [this forum thread](#) for details and illustrations. Note that each model has a different number **#nn**: #1 #2 etc. for all properties.

Example lamp entry – light source

```
id#2=okh#Lamp_wall
name#2=Lamp wall
tags#2=Lamp with light source
category#2=Lights
icon#2=/Lamp_wall_plain.png
model#2=/Lamp_wall_plain/lamp_plain.obj (link to model file - inside library)
multiPartModel#2=true
width#2=22
depth#2=27.5
height#2=34
movable#2=true
doorOrWindow#2=false
lightSourceX#2=11 (light source distance into model X-axis)
lightSourceY#2=11 (light source distance into model Y-axis)
lightSourceZ#2=27 (light source distance into model Z-axis - up)
lightSourceColor#2=#ffff33 (HTML: HEX triplet RGB: #ff0000=red #ffff00=yellow etc)
lightSourceDiameter#2=10
elevation#2=120
creator#2=Ola-Kristian HOFF
```

Using red-tones to lower "colour temperature" (for instance for candles or fireplaces) works well. However, not only rendering but also how the mind perceives lower colour temperatures may make it difficult to achieve accurate results. Placing the light source accurately is important as the model surrounding the light source will work just like a normal lamp to shield and direct the light in the correct direction.

Information, legal and copyright notice

Sweet Home 3D (SH3D) is presented as home decoration software, but has grown into more than that. It allows you to design a full home and decorate it using furniture. Furthermore it lets you change colours and textures of all elements. And it works on Linux, Mac and Windows. You will find more on <http://www.sweethome3d.com/>. Mr Emmanuel PUYBARET is a Java developer and the creator of Sweet Home 3D. For licenses and contributions, see the [website](#) and the About notices in SH3D.

SVG is a royalty-free graphics format developed and maintained by the W3C SVG Working Group, see <http://www.w3.org/Graphics/SVG/> for details.

Models (furniture, including staircases, windows and doors) for your SH3D design you can find on the Internet (SH3D supports several formats such as .obj files and .dae files, zipped if you like). Beware of copyright limitations. If you find a lovely model online, that does not mean you can distribute (or even share) it with your home. If you share a home with models found online, you expose yourself to copyright liability. Models distributed with SH3D and from the SH3D website have liberal copyright licenses and can be distributed (but if you use your home publicly, especially commercially, double-check and always credit the designer of the model and the software you have used). The same restrictions, to some extent, can even apply to textures. Home drawings you find online can be an inspiration, but if you copy an elaborate home, the architect may not be too pleased. In short, if you find something clever you want to use, do not pass it on (as your own) without prior permission, and even if such permission is given, credit the designer.

If you submit a model to SH3D (use sourceforge), make sure you comply with the copyright requirements and consider using a library if you wish to include all your advanced properties.

Drawing a home with SH3D is relatively easy. But if you wish to build the design, you should seek professional building advice. The outline of your home is not even half the job an architect needs to do. For instance, you may have to satisfy energy requirements that are not straightforward to calculate. Often there will also be standards you need to satisfy: for instance, if you use standardised insulation layers for an outer wall on a wooden home, the wall thickness is given (e.g. some 28cm, as opposed to old building standards where you could get away with 15cm thickness). The builder will also need accurate instructions on the elements of the house, have someone figure out the structural strength of the construction, etc, etc.

Me, I am but a grateful, amateur user of SH3D and I have absolutely no connection with SH3D, SVG or any of the software or tools mentioned. This guide was originally just a set of notes for myself when I needed to understand how to customise my furniture for a modest redecoration project. When I share this document, it is but a vain attempt to give something back because SH3D has provided me with great help in visualising my construction thoughts. And who knows, maybe this guide could even save someone the trouble of figuring it all out for themselves. Thanks to Emmanuel PUYBARET and Hans DIRKSE who took the time to read my draft and provide me with guidance. All mistakes are mine.

Use this document freely for your development. Formally, it is released under [CC-BY](#) license, it is intended only as a quick-guide for the users to figure out how to use SVG with SH3D FurnitureLibraryEditor and advanced properties editing. I take *absolutely no responsibility for errors, omissions or other mishaps*.

I welcome comments (well, usually), preferably in (or via) the SH3D forum.

Ola-Kristian HOFF

December 2013

References

Personal link file for working with SH3D, SVG related links on top.

SVG references and tutorials

W3C Scalable Vector Graphics	http://www.w3.org/TR/SVG/
SVGBasics	http://www.svgbasics.com/
SVG tutorial at Illinois State University	http://www.itk.ilstu.edu/faculty/javila/SVG/
Mozilla SVG tutorial	https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial

SH3D links for SVG and catalogue properties

Forum introduction to simple SVG shapes	http://www.sweethome3d.com/support/forum/viewthread_thread.1550#12689
Forum: how to add advanced window properties.	http://www.sweethome3d.com/support/forum/viewthread_thread.2006#8919
DefaultFurnitureCatalog.properties (file with furniture properties, will show examples of how to add lightsource and door/window properties)	http://sweethome3d.cvs.sf.net/viewvc/sweethome3d/SweetHome3D/src/com/eteks/sweethome3d/io/DefaultFurnitureCatalog.properties?view=markup
Properties catalogue for SH3D	http://www.sweethome3d.com/javadoc/com/eteks/sweethome3d/io/DefaultFurnitureCatalog.PropertyKey.html

Terminology and colours

Merriam Webster Visual dictionary (useful for finding English names for house elements)	http://visual.merriam-webster.com/
Material names recognised by SH3D	http://www.sweethome3d.com/images/Java3DCubeColorsWithTransparency.png
HTML basic named colours	http://en.wikipedia.org/wiki/HTML_color_names#HTML_color_names
RAL colours - overview	http://ralcolors.net/

Legal links

Eurocodes	http://eurocodes.jrc.ec.europa.eu/
DIRECTIVE 2002/91/EC:energy performance of buildings	http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32012R0244:EN:NOT
Directive 89/106/EEC relating to construction products	http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31989L0106:en:NOT
PBL	http://www.lovdato.no/dokument/NL/lov/2008-06-27-71
TEK 10, byggeteknisk forskrift	http://www.lovdato.no/cgi-wif/ldles?doc=/sf/sf/sf-20100326-0489.html
Byggsak	http://www.lovdato.no/cgi-wif/ldles?doc=/sf/sf/sf-20100326-0488.html
Eurokoder	http://www.standard.no/no/Fagomrader/Bygg-og-anlegg/Eurokoder/Samlet-oversikt-over-Eurokodene/