

# Swap Curve Type

The Swap tool changes the definition of an existing FredoSpline curve based on its control points

## ► Swapping curves

### With Preselection

- Launch the Swap tool with one or several FredoSpline curves preselected
- The curves will be shown with the swapping transformation
- You have the opportunity to **change the curve and modify curve parameters**
- **Click anywhere** to perform the swap
- After swapping, you can still change the curve and parameters and use variators

### Interactive (no preselection)

- Launch the Swap tool with no FredoSpline curves preselected
- You have the opportunity to **change the curve and modify curve parameters**
- **Hover** over a curve
- The curve will be shown with the **swapping transformation** in purple
- **Click** to perform the swap
- When done, you can still **adjust the parameters**
- You can hover over **other curves** and apply the same method
- **Click in empty space** to exit

### To finish (preselection or interactive)



- **Click in empty space** to **exit** the Swap tool
- **Click** on a swapped curve to **switch to the Edition tool**
- **Alt-Click** anywhere to **pursue** interactive swapping of other curves

## ► Show / Hide curve vertices

You can display or hide the vertices of the curves

This option is available for all tools

### To show / hide vertices...

- From the option bar  

- From the toggle buttons  

- From the **VCB**: type **\*** to toggle display
- From the contextual menu

## ► Flow Management


### Pause

- **TAB** to pause: this interrupts the input or current operation




- This is useful to change the curve parameters in the palette
- **TAB** again to resume


### Undo

- **Esc** or **Ctrl-Z** or  to **undo** the last operation

## Redo

- **Esc** or **Ctrl-Y** or  to **redo** the last operation

## Finish and Exit

- Click on  to exit
- or **Click in empty space** in most situations (as shown by the cursor)
- For Creation tool, you have **additional finish options** (check the contextual menu)

## Default Parameters

- Click on  to access the **Default Parameters dialog**

## Palette buttons

- You can also use the **palette buttons**



## ► Variators

At any time, you can modify the numeric parameters of the curve via **Click-Drag**

Variators are available for all FredoSpline tools

### In Palette button

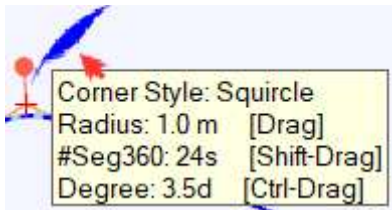
- **Click-Drag** in the button → Variator will appear and can be moved

### In empty space

- **Click-Drag** in the button → Variator will appear

## Modifiers

- When there are **several parameters**, press-down a modifier key (**Ctrl**, **Shift**, **Alt**) while dragging
- The applicable modifier for a parameter is indicated in the **tooltip**




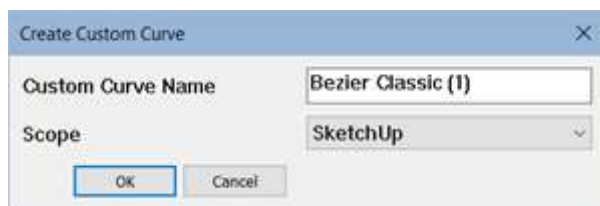
## ► Custom curves



FredoSpline comes with a set of curves with predefined parameters

You can however create Custom curves with **your own parameters**


### Creating a Custom curve

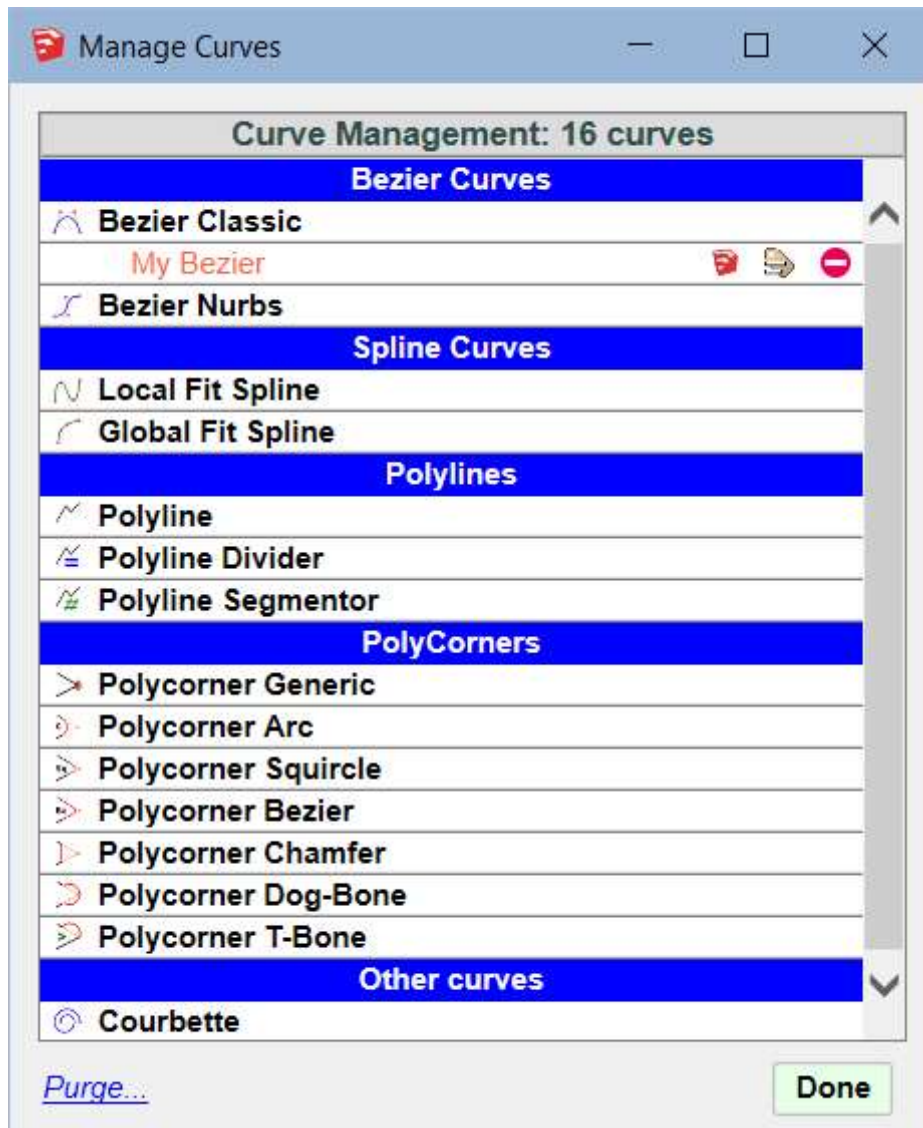
- At any point in time press the button  to create a custom curve with the current type and parameters
  - Note that the parameters must be **different** from the default parameter of the curve type
  - This is indicated by a **small \*** appended to the curve name
- You will be prompted for a **name** and the **scope**



-  **SketchUp:** in All models and All SketchUp versions
-  **Model:** only in the model and its copies


## Managing Custom curves


- Press the button  (next to the curve name) to manage all your Custom curves
- A dialog box appears; you can **rename**, **delete** or **change the scope** of your Custom curves



### ► Curve Family: BEZIER

Smooth curves *NOT* passing by the control points

 **Classic Bezier** is of degree 3 and tangent to the control polygon

 **Bezier Nurbs** can have a higher degree (**tension**) giving it more rigidity

All Bezier curves support **Cusp** (discontinuity at control point) and **native loop**

#### Global Parameters (Classic and Nurbs)

- **Number of segments:** (0 for automatic)

#### Global Parameters (Nurbs only)

- **Tension** [0..100%]: increase the degree to gives more rigidity


#### Local Parameters

- **Weight** [0..1] and [ 1..10]: proximity to the control point (Default 1)

### ► Curve Family: SPLINE

Smooth curves passing *BY* the control points

 **Local Fit Spline** is close to the control polygon and has many local tweaking

 **Global Fit Spline** passes very close to control points and keeps a global shape

All Spline curves support **Cusp** (discontinuity at control point) and **native loop**

### Parameters for Local Fit Spline

- **Smooth Angle:** deviation angle for smoothing the curve (in degree)
- **Tension** [0..100%]: Kind of rigidity making the curving closer or farther to the control polygon
- The tension can be adjusted locally at each control point
- **Tangent** orientation at control points, symmetrically and asymmetrically

### Parameters for Global Fit Spline

- **Number of segments:** (0 for automatic)
- There are NO local parameters


## ► Curve Family: POLYLINE

Polylines based on the control points with division options

 **Polyline Basic:** Simplest polyline just joining the control points

 **Polyline Divider:** Polyline divided in segments of specified length

 **Polyline Segmentor:** Polyline divided in a specified number of segments

 **Polyline Path:** Polyline divided for camera path frames

When divided, polylines may not pass through all control points

Polylines do not have local parameters

Polylines are usually used to divide a path formed by an edge sequence

### Parameters for Divider

- **Interval:** length of segments
- **Method:** Specify how to handle the division


### Parameters for Segmentor

- **Number of segments:** Number of equal segments
- **Method:** Specify how to handle the division and the respect of the control points


## ► Curve Family: POLYCORNER


Polylines with **corner shaping at vertices**, configurable **globally** and **locally**

 **Polycorner Generic:** Fully configurable Polyline, with any corner type


 **Polycorner Arc:** Corners are based on **Arc of Circle**

 **Polycorner Squircle:** Corners are based on **Squircle**

 **Polycorner Bezier:** Corners are based on a **Bezier junction**

 **Polycorner Chamfer:** Corners are based on a **straight Chamfer**

 **Polycorner Dog-Bone:** Corners are based on a **Dog-Bone**

 **Polycorner T-Bone:** Corners are based on a **T-Bone** on the longest side

You can mix corner types for any Polycorners, and remove corner shaping at any vertice

All Polycorner curves support native loop

Each type of corner has its own parameters, available globally and locally