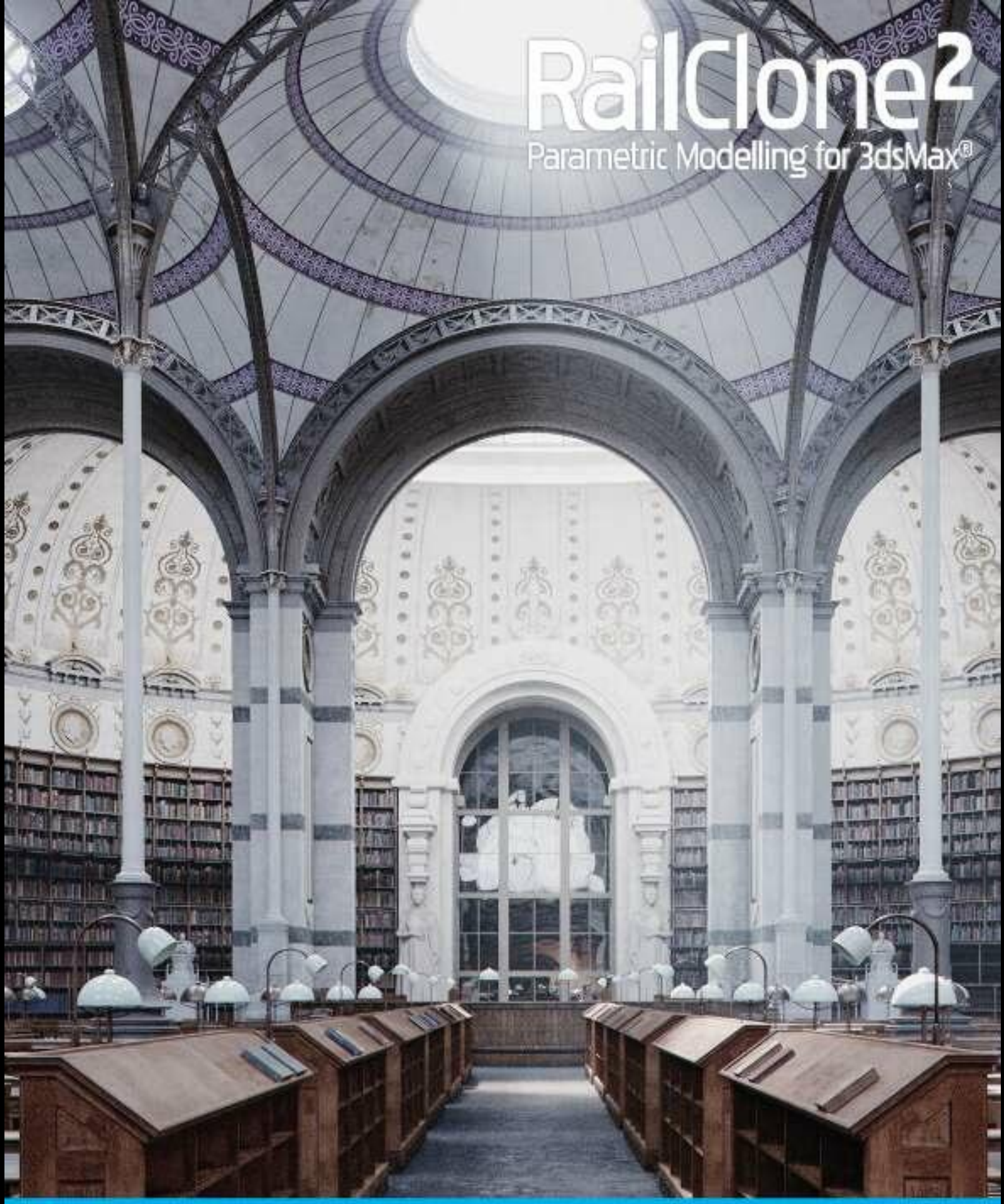
RailClone 2

Parametric Modelling for 3dsMax®



iToo software

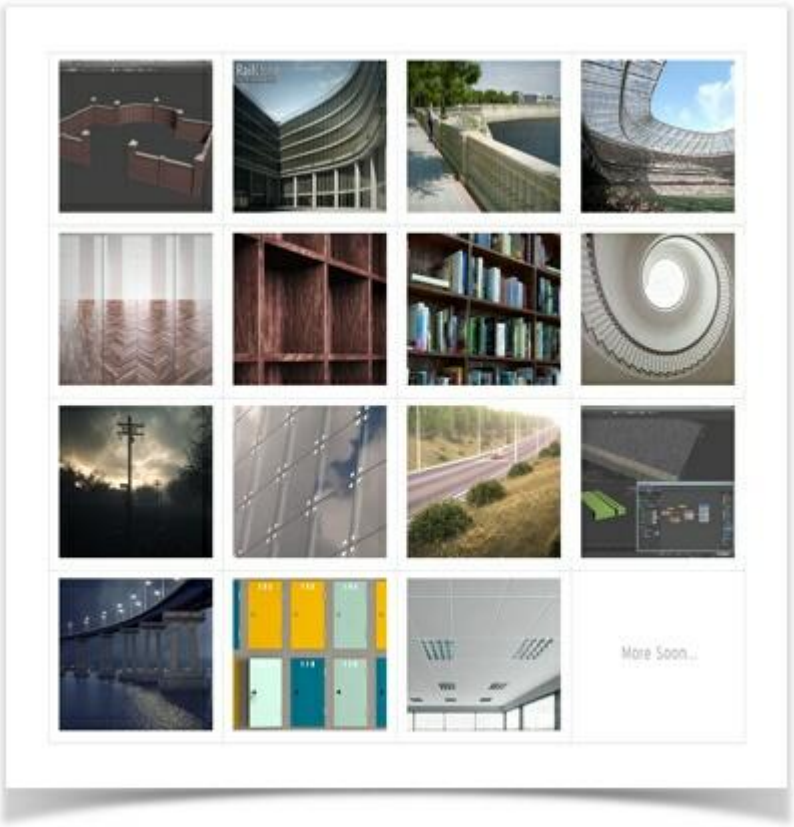
Documentation

Image by Bertrand Benoit

Table of Contents

If you're new to RailClone and not sure about where to begin, you've come to the right place! This introductory guide has been written to help you get started creating powerful parametric models in next to no time. Follow the chapters below to learn how to get the most out of RailClone.

Chapter 1 - Introduction



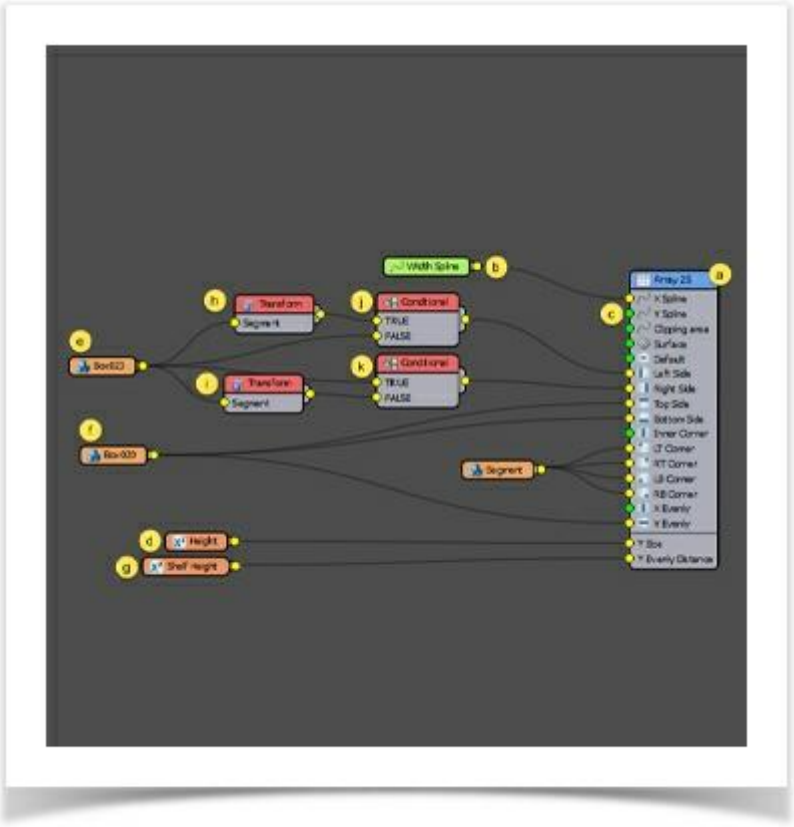
Chapter 2 - Learning RailClone and getting help

Chapter 3 - RailClone versions: Lite and Pro

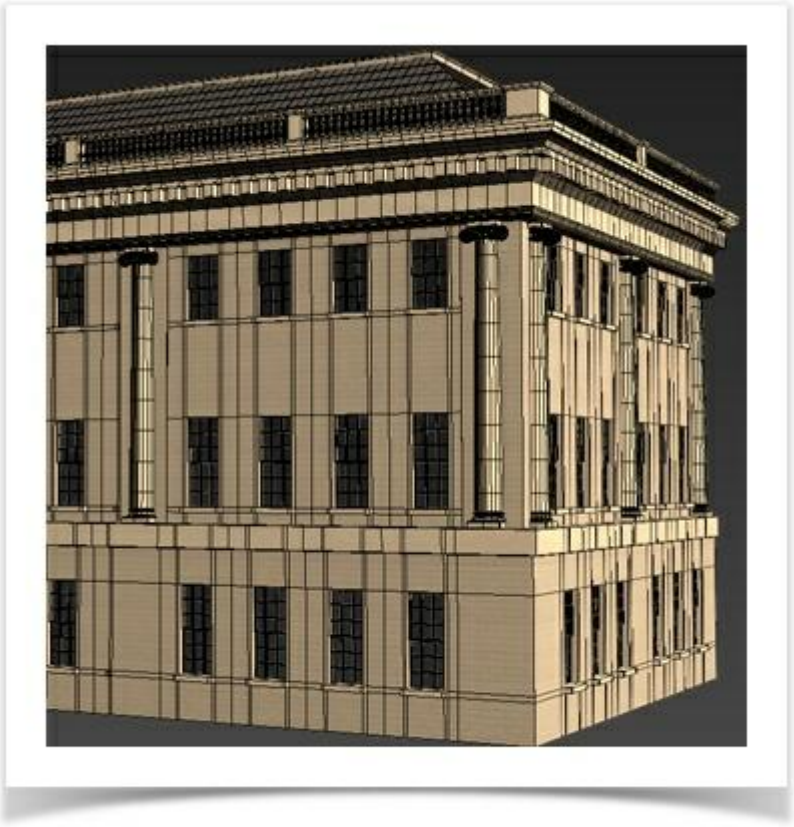
Chapter 4 - Key concepts: Array based instancing

Chapter 5 - The RailClone interface

Chapter 6 - How to use the library

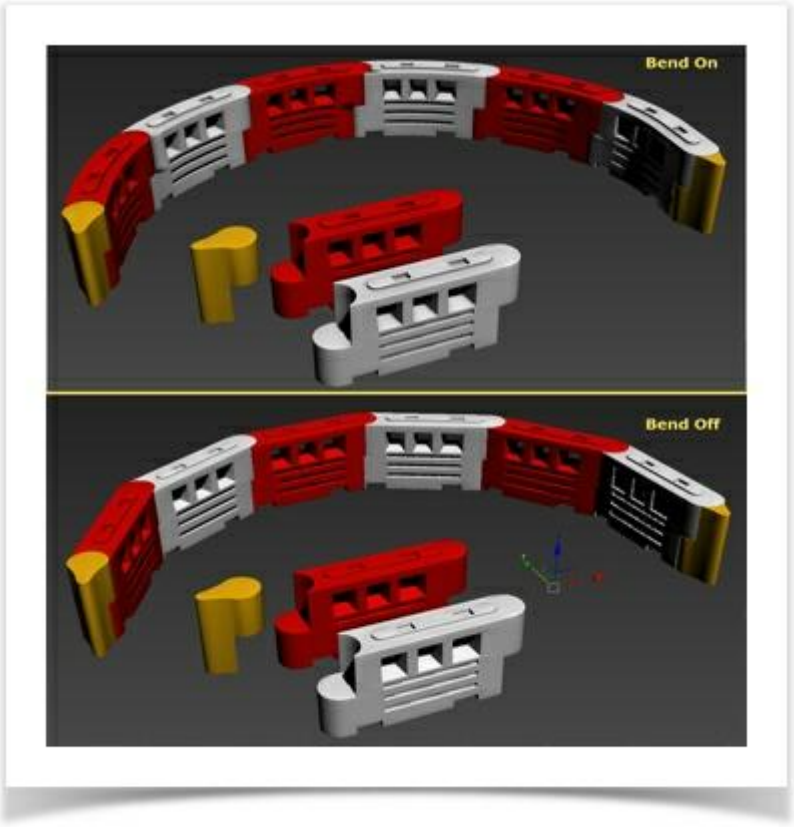


Chapter 7- How to edit a RailClone style



Chapter 8 - Create your first 1D array

Chapter 9 -Create your first 2D array



Chapter 10 - Techniques to deform geometry

Chapter 11 - How to overlap and transform geometry

Chapter 12 - How to combine, select and sequence geometry

Chapter 13 - Next Steps

1 Introduction

Congratulations on downloading RailClone, the artist-friendly procedural modelling and instancing plug-in for 3ds Max. RailClone has a unique approach to parametric modelling that is powerful, fast and easy-to-learn. To help you get started, we've written this Beginners Guide to introduce the core concepts and get you up and running with RailClone as quickly as possible. It is our intention that by the time you reach the end of this document, you will be ready to make RailClone an indispensable part of your your day-to-day workflow.

1.1 Using this guide

We recommend working through this guide from start to finish. It has been structured to that each new section builds on the concepts learned in the last chapter. To help you get more from the guide, each new concept features the following sections.

Exercises

The best way to learn RailClone is use it. we've included exercise throughout the guide to help you pick up the foundational skills required to use RailClone for many of the relevant day-to-day tasks you might encounter . We recommend making the effort to go through these exercises, but we're aware that time is precious, so the exercises have deliberately been kept short and simple.

You can download the source files for the exercise below. The .max files are in 2010 format with standard materials.

Getting Started Exercise Files

Relevant Tutorials

At the end of each chapter you will find links to recommended tutorials. Follow these to learn more about each subject, see the principles put to use and learn more advanced skills

Links to documentation

To find out more about a subject, each chapter provides links to the relevant areas of RailClone's online documentation. If you have a question, there's a good chance you'll find the answer there.

1.2 Requirements

No special skills, renderer specific knowledge, programming experience, or additional software is required to use and learn RailClone. If you can find your way around 3DS Max then you're ready to go!

2 Learning RailClone and Getting Help

In addition to this guide, we have many other channels available to help you learn our plugins, talk to other users, and get help when needed. Below are some of the resources you may find useful as you start working with RailClone. Click on the image to access the resource.

2.1 Documentation

The RailClone User Guide provides comprehensive descriptions and commonly used procedures for every setting in RailClone. If you have a question, you'll often find the answer here.

To access the documentation from within the plugin:

1. Select the RailClone object.
2. In the Modify panel, open the **General** rollout.
3. Click on

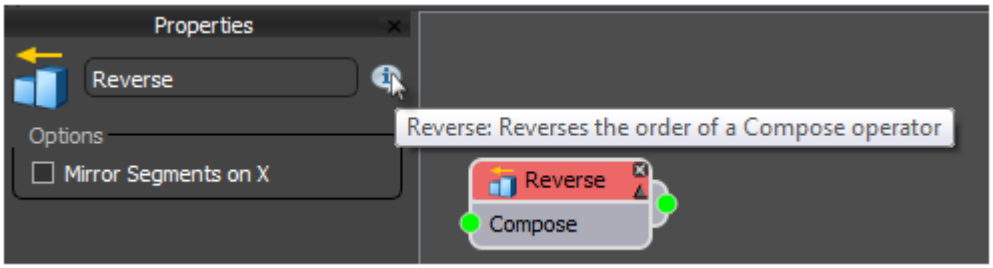
   to open the online help in your default web browser.

Alternatively if you need to read the documentation offline, we also have PDF and EPUB versions available.

Tooltips

Many buttons in RailClone display additional information if you hover the mouse over them. In the style editor, watch out for this icon

which will reveal more information about the purpose of a node.

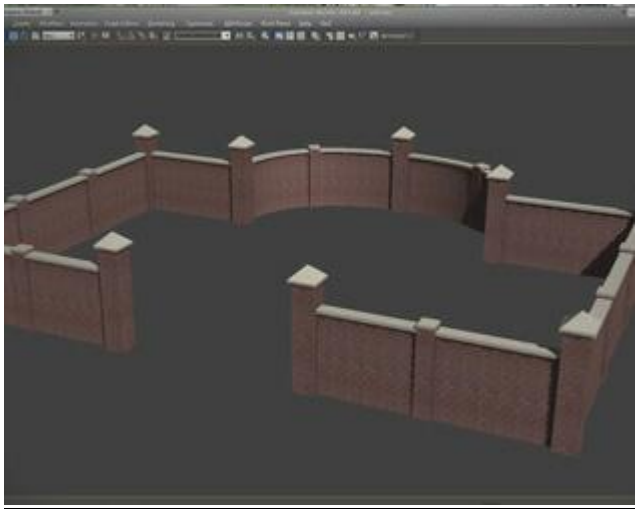


2.2 Tutorials

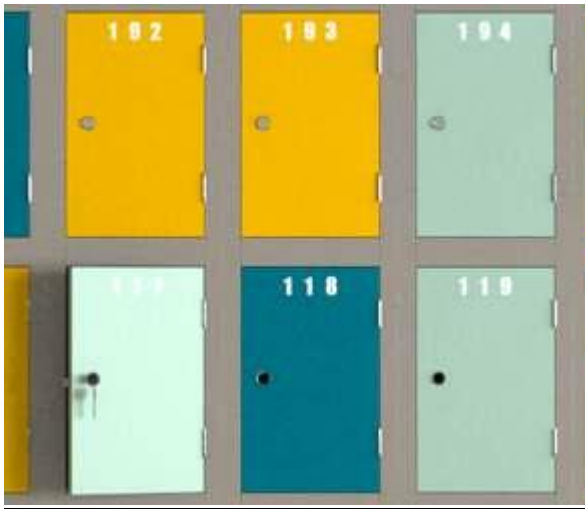
We regularly release Tips and Tricks tutorials to help you learn new techniques for RailClone and stay up to date with recently introduced features. In particular we have 4 tutorials that may be helpful for new users. The Masonry Wall tutorial introduces the basic concepts of RailClone and teaches you how to create a brick wall using a 1d array. Following that, the Curtain Wall tutorial (

) introduces 2d arrays using the example of a double curved façade. Having learned the basics, both 1D and 2D arrays are combined in the Seaside Promenade and Stadium tutorials that also introduce many of the most commonly used operators, and demonstrate RailClone's ability to handle huge amounts of geometry by automatically instancing objects.

You'll find these, plus more tips and and tricks videos in our tutorials section. Some are these are shown in the thumbnails below, but the section is often updated so check regularly!




RailClone



2.3 Forum

Our forums are a great place to get you questions answered, share your experiences with other users, and learn new tips. I f you own the Pro version of RailClone, you'll get premium support on the restricted boards. Our technical support staff monitor the forums closely so you can be sure of a timely response. To access the Pro boards, use the username and password you received with your license. If you can't remember your login, just drop us a line requesting that we resend it.

2.4 Scenes

We periodically release complete scenes created using Forest Pack and RailClone. These can be useful learning aids, allowing you to reverse engineer many useful styles. Get them from our Downloads page along with the scene files for all the tutorials.

2.5 Helpdesk

If you have a question that is not answered in any of the channels above, you can contact our technical support staff directly. To do this please use this contact form.

> **Business Hours**
>
> Our business hours are from 9 a.m. to 3 p.m. and 4 p.m. to 6 p.m. (GMT +1)

2.6 Newsletter and social media.

If you're a Pro user you'll be kept up to date with all iToo related news including product releases, updates, tutorials and more, via our regular newsletter. In addition all users are welcome to follow us on Facebook, Twitter, Vimeo, and YouTube.

3 RailClone Versions: Lite and Pro

RailClone comes if two different flavours. The **Lite** version is completely free to download and use, even commercially, but includes some restrictions as outlined in the table below. The **Pro** version is completely unrestricted, including the full feature set, and also includes a much larger styles library and comes with RailClone Tools. This guide serves as a useful primer for RailClone, irrespective of whether you're a Lite or a Pro user.

3.1 Differences between Lite and Pro

Lite	Pro
Limited to a single generator	**Unlimited** generators
Limited to flat splines	Includes **3 different adaptive height modes** to deform geometry on the Z axis
Limited to 3 segments	**Unlimited segments**
Cannot be converted to an editable mesh	Can be converted to an **editable mesh**
Reduced style library	**Full style library** included
Library cannot be customised	**Library** is **fully customisable**
RailClone Tools not included	RailClone Tools included.

Aside from these restriction both versions are identical in appearance and behaviour. Throughout this guide, where an exercise, note or procedure is only relevant for Pro users you'll see

PRO only

displayed.

4 Key Concepts: Array Based Instancing

4.1

4.2 What is RailClone?

RailClone is designed to help artists create sophisticated reusable parametric models using an easy to understand node based editor. At the heart of RailClone are two array-based **Generators** . These use rules to combine, transform, deform, slice, bevel, UV map, and distribute meshes known as **Segments.** Once you understand how generators work, creating your own unique RailClone objects is simple and fun!

Compared to many other parametric modelling tools, RailClone works at a much higher level of abstraction. RailClone isn't a visual programming tool, it's a sophisticated array based modeller with tremendous Instancing power. There's no need to create complex algorithms or programmatically generate meshes from scratch, instead RailClone uses modular pieces of geometry created using standard modelling tools as its building blocks. These are assigned to different parts of an **Array** and manipulated based on the the instructions contained **in rule-sets**. We call the combination of rules, arrays and segments a **Style.**

For example, the image below illustrates a rope barrier style created with RailClone.

This style is created using only two **Segments**:

A post....

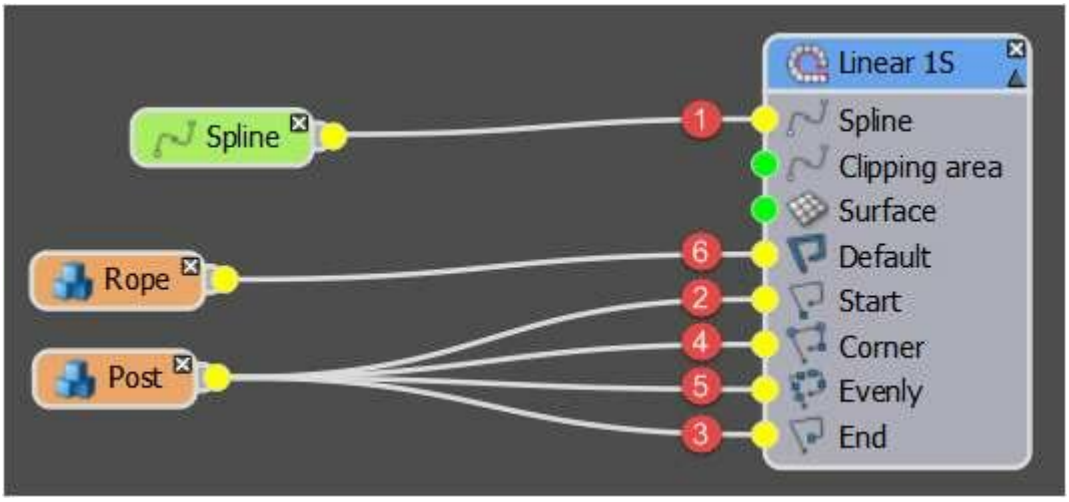
...and a rope.

These two segments are organised into a one dimensional array by a Generator, which uses a simple rule-set. If we were to write out this rule-set in English it would read as follows:

1. Create a **1 dimensional** array deformed along a **Spline** path.
2. At the **Start** of the spline, add a post.
3. At the **End** of the spline, add a post.

4. If there is a **Corner** vertex, add a post.
5. Along the length of the spline, add **Evenly** spaced posts every 1.2m.
6. Between all these posts, add a rope and scale it so the it fits exactly between the posts.

In RailClone, using the node based **Style Editor**, the rule-set looks like this:



As you can see this array has 5 parts: the start, end, corner, evenly and default. Each of these parts is represented by an **Input** on the generator's node. To add geometry to the array, it is only necessary to wire a **Segment** to an input.

This example contains the 3 essential ingredients to create a RailClone object, a **Generator** (rule-set), **Base object** (spline) and at least one **Segment** (geometry). In the node tree shown above, the *Linear 1S* node is a type of **Generator** which creates the array, the *Spline* node is a **Base Object** and determines the path, and the *Rope* and the *Post* nodes are **Segments** that add the geometry. Let's look at these 3 components in more detail.

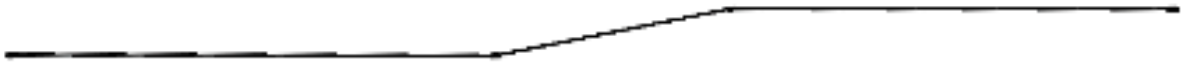
4.3 Generators: RailClone's two array types

To create parametric objects with RailClone you compile geometry using one and two dimensional arrays. Each of these two arrays is created using a different generator, and each has different inputs. The L1S generator is simplest of the two with 5 inputs. This generator is the foundation of RailClone's approach to modelling.

RailClone's second generator is designed to create 2d arrays and is more sophisticated with a total of 12 possible inputs. If you stack several L1S arrays on top of one another you get a 2d array and this, in essence, is how the A2S generator works. Because of this relationship between the two generators, understanding the L1S generator is the most important step in learning RailClone.

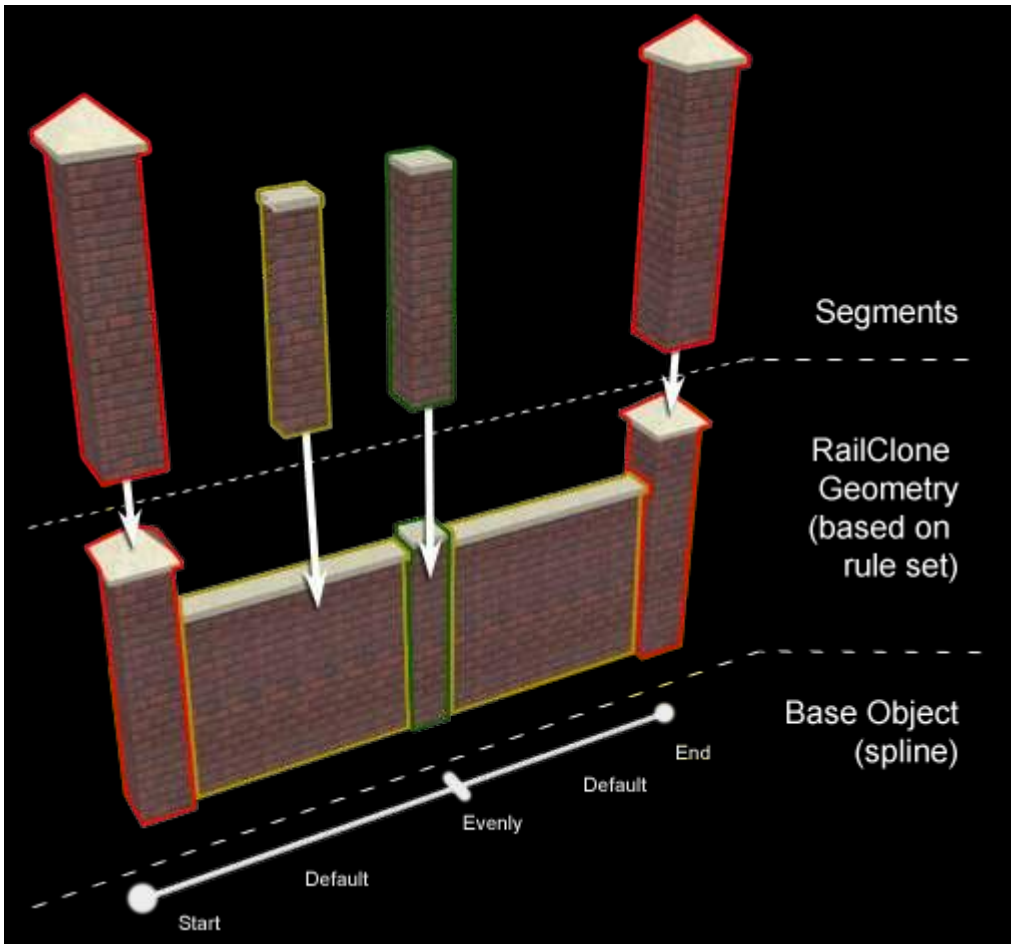
The one dimensional or L1S array

As we've mentioned the L1S array has 5 inputs. Each of these relates to a position on the X Axis of the array and/or the position on the base spline. To create parametric objects with RailClone, it's essential to understand how these 5 elements interact. The following animated image and descriptions illustrate the inputs used by the L1S Generator and their position on a base spline. Though it looks simple, it's a deceptively powerful system that can generate a limitless number of different styles.



1. **Start:** This input is used to position a segment at the start of the array.
2. **End:** This input is used to position a segment at the end of the array.
3. **Corner:** This input is used to add a segment at each of the intermediary vertices of the spline. You can set this so that it only creates a segment on certain types of vertex i.e. corner. bezier-corner, bezier, smooth, or if you prefer, all of these.
4. **X Evenly:** This input is used to create evenly spaced segments along the spline. The distance between them is fully adjustable and there are several algorithms available to calculate the spacing.
5. **Default:** This is used to fill in the remainder of the array and in the event that the other nodes have no inputs. If an input is left empty, the segment wired to default will be used in its place.

To see this in action, examine this image from our Masonry Wall tutorial. It illustrates the segments used by the style and the input slots to which they're attached.

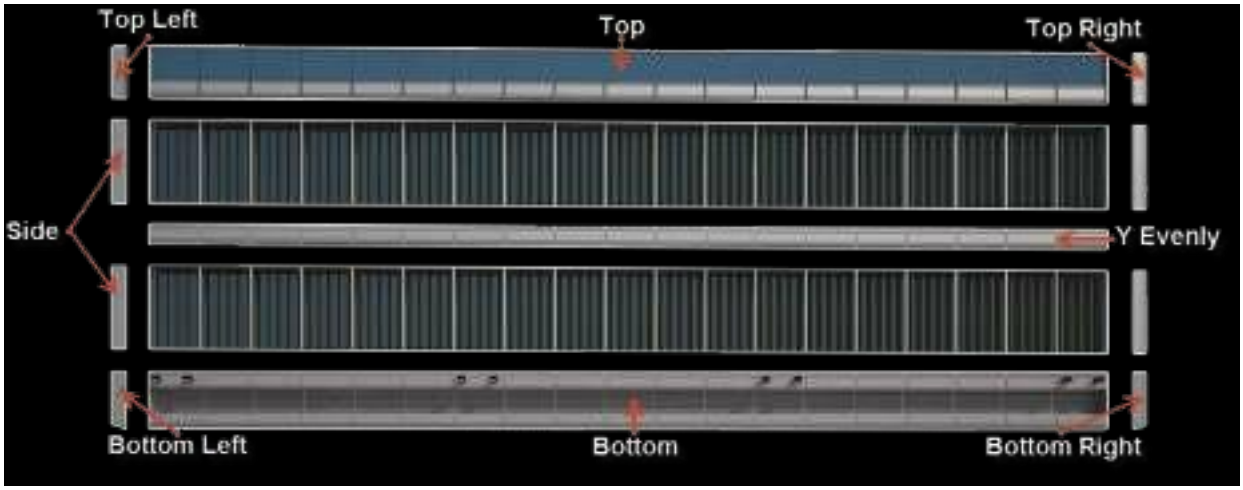


The two dimensional or A2S Array

Using the A2S Generator, segments can be positioned and deformed in a two dimensional grid with separate target slots for top, bottom, sides, each corner and evenly spaced rows and columns. This powerful generator is great for walls, pavements, façades, roofs and many other advanced structures. The image and descriptions below illustrate the inputs used by the A2S Generator, and their position in the array.

1. **Left Bottom Corner:** Use this input to position a segment in the bottom left hand corner the array.
2. **Right Bottom Corner.** Use this input to position a segment in the bottom right hand corner the array.
3. **Left Top Corner:** Use this input to position a segment the the top left hand corner the array.
4. **Right Top Corner:** Use this input to position a segment in the top right hand corner the array.
5. **Left Side:** Use this input to add a column of segments on the left hand side of the array.
6. **Right Side:** Use this input to add column of segments on the right hand side of the array.
7. **Inner Corner:** Use this input to add a column of segments at each of the intermediary vertices of the construction spline.
8. **Bottom Side:** Use this input to creates segments in the bottom row of the array
9. **Top Side:** Use this input to creates segments in the top row of the array
10. **X Evenly:** Use this input to creates evenly spaced columns of segment.
11. **Y Evenly:** Use this input to create evenly spaced rows of segments.
12. **Default:** Use this input to assign a segment that is used to fill in the remainder of the array and in the event that the other inputs are empty.

The following image is from our Curtain Wall tutorial and illustrates how the relationship between the segment's geometry and the input slots described above creates a working style. Note that although this generator has 12 input slots, it isn't necessary to use them all. In this example only the corners, sides, top, bottom, Y Evenly and default have been used. Inner Corner and X Evenly were not required to create this style. When an input is not used the generator is able to intelligently reconfigure itself to use only the inputs that are active.

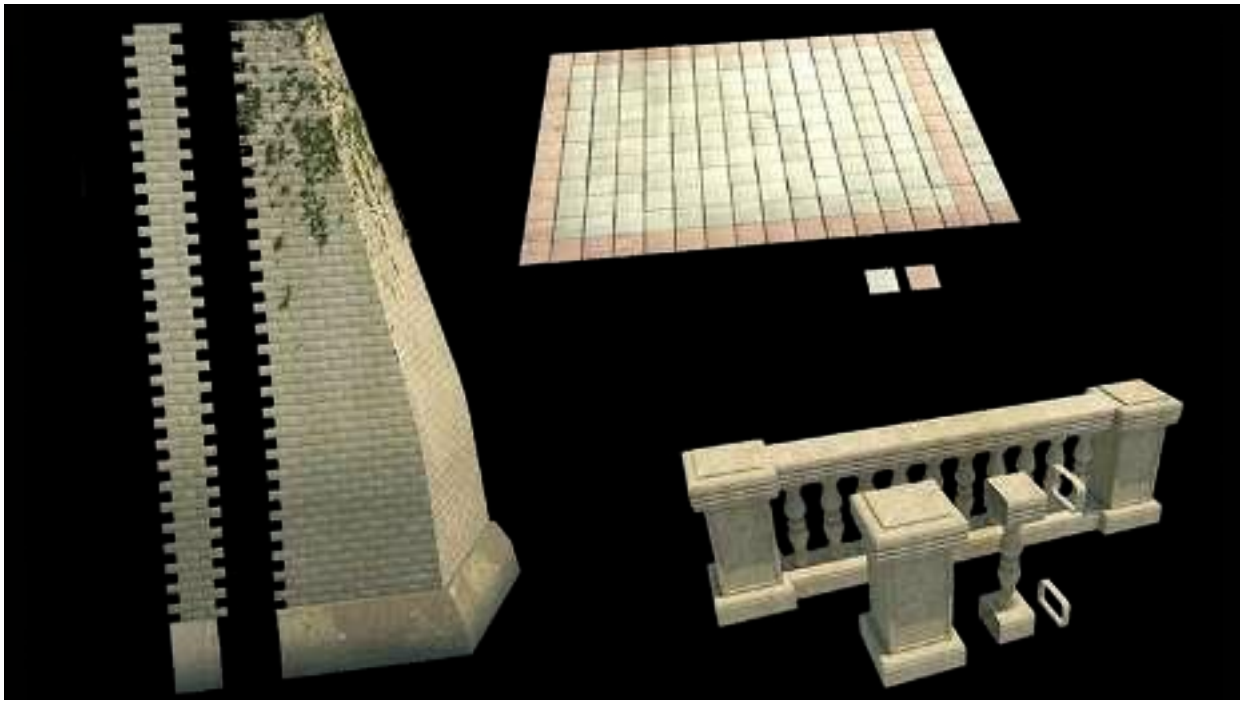


The inputs used for a curtain wall style

Render of the curtain wall style

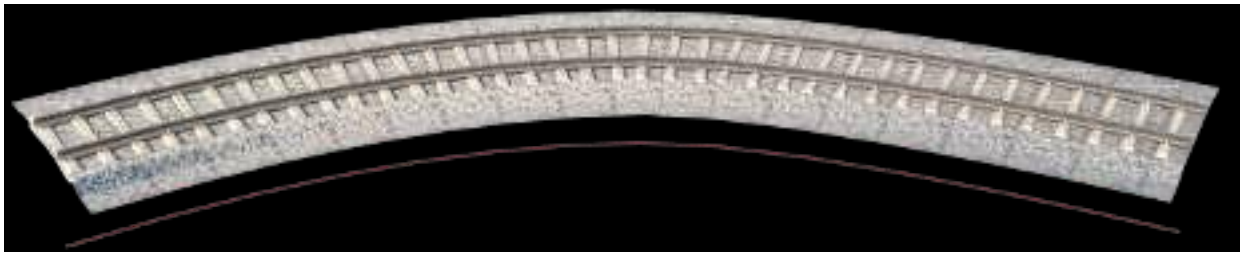
4.4 Segments

Creating new styles with RailClone necessitates thinking about your models as repeatable parts of a jigsaw, that slot together into a 1D or 2D array. We call these jigsaw pieces **Segments**, they are the individual building-blocks, duplicated and combined by the generators, to create a final model. A segment could for example be a single complete object distributed along a path (for example, a lamp-post), or a modular set of parts that snap together to create a precise unified mesh (for example, a space frame). The geometry used for Segments is typically very specific to each style, but the image below show 3 different RailClone objects from the Seaside Promenade tutorial with their respective segments.

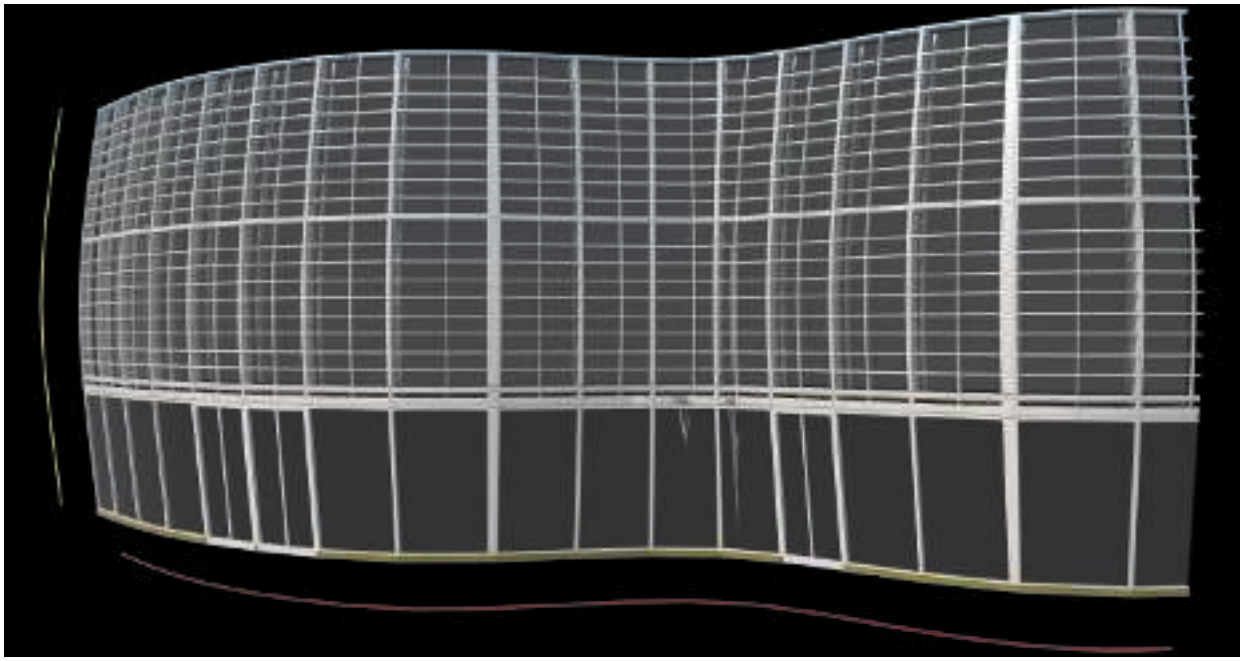


4.5 Base Objects

To set the size of an array you have two options, either use scene units to specify the length and height, or use splines as paths for geometry to follow. These paths are referred to as **Base Objects**. Depending on the type of generator used you can employ up to 3 splines. For example this Railway preset from the included library uses the spline marked in red to deform the track.



When using A2S generators (two dimensional arrays), two splines can be used, one for the X Axis and another for the Y. In the curtain wall example below, the red spline governs the path while the yellow determines the height. Both splines can be curved to generate geometry that would be time consuming to create using other tools or techniques.



In addition to defining the axes of a generator, the third type of spline base objects functions as a clipping area. These splines must be closed and RailClone uses Max's boolean functions to include or exclude geometry by slicing segments. This will be discussed in more detail in the Intermediate guide.

4.6 Operators

Operators allow you to perform all kinds of modifications to segments. These nodes allow you to transform, mirror, combine, and sequence segments; randomise geometry, material IDs or UVW coordinates; combine segments to form a new composite object; create conditional rules and much more.

4.7 Parameters

Finally, **Parameters** allow you to link the properties of generators, segments and operators to one-another or to numerical nodes. Use parameters to link different parts of a RailClone style together and make adjusting multiple objects from a single input a cinch. Mathematical relationships can be added between properties by using Arithmetic operators and for the more adventurous you can even write custom expressions.

4.8 Instancing

RailClone shares the instancing power of Forest Pack Pro. It has been designed from the outset to create huge amounts of geometry using popular rendering engines. Any segment that is not deformed or sliced is instanced automatically, saving memory and improving render-times.This happens automatically behind the scenes, though as a general rule the more you minimise deformation and slicing, the more optimised your model will become.

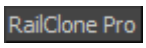
For an example of instancing, take a look at the stadium scene (show below) from our 5 part tutorial series. This scene uses RailClone to create all the seating (and almost everything else). Each seat model has **14,018** polygons and the stands contain approximately **50,000** chairs. This gives us a total of **700,900,000** polygons, and that's before adding the crowds, flags, roof, and grass!

5 The anatomy of a style - RailClone's approach to modelling

Before you start using the included library or modifying and creating your own styles, it's useful to take a few moments to get acquainted with the user interface. To explore the UI, first create a new RailClone object by following these steps:

1. Go to 3ds Max's **Command** > **Create** > **Geometry** panel and change the category to **iToo Software**.

2. Click



.

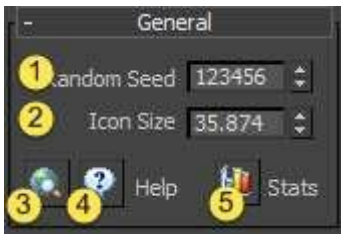
3. Click and drag in the **Top** or **Perspective** viewport to create a new RailClone object.
4. Switch to the **Modify Panel** to access the tools required to load a library preset or create new style.

5.1 RailClone's 3 UI Components

The starting point when using RailClone is always the **Modify panel**. From here you can adjust many of the object's global settings, open the **Library Browser** to load and organise presets, and launch the **Style Editor** to modify existing styles and create new ones. Let's start by taking a look at the layout of the modify panel.

1 - Modify Panel

In addition to the **About** rollout, RailClone's Modify panel UI consists of 5 rollouts, shown below.



The General rollout

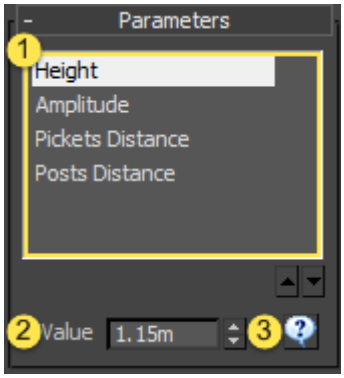
The General rollout is used to adjust global plug-in settings and provides access to statistics, help files, and updates.

1. **Random Seed** Sets the seed value used when generating random properties in a style.
2. **Icon Size** Adjusts the size of the icon used when no geometry is displayed. This is purely cosmetic and has no influence on rendered geometry.
3. **Check for updates**. Checks your plug-in is up-to-date.
4. **Help** Open the online documentation in your default browser.
5. **Stats** Shows useful information about the current style, including build time, number of segments and error messages.

The Style rollout

The Style rollout is used to load, design and control rule-sets used by the RailClone object to build a parametric model. As well as being able to edit many attributes common to all styles, in this rollout you can load presets using the built in **Library Browser,** or create and edit Styles using the **Style Editor**.

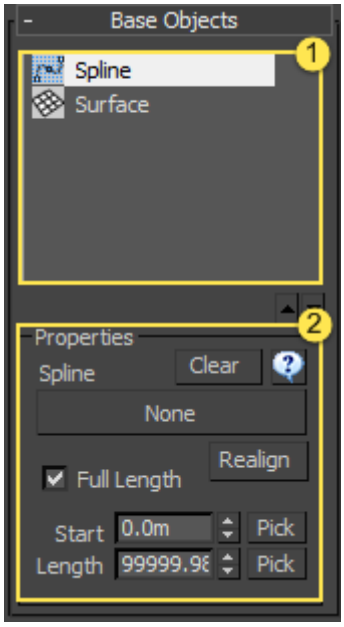
1. **Preview Image** Displays a preview image when a style is loaded from the Library.
2. **Style Name** Click to open the Style Library. The button's text displays the name of the current style.
3. **Style Description** Click to read a description of the currently style, including a list of the parameters and base objects used.
4. **Style Editor** Click this button to open the Style Editor, used to create or modify styles.
5. **Copy Styles** These 3 buttons are used to copy the current style **to** other RailClone objects or copy a style **from** another RailClone object
6. **Geometry** This group contains editable parameters that are common for all styles.



The Parameters rollout

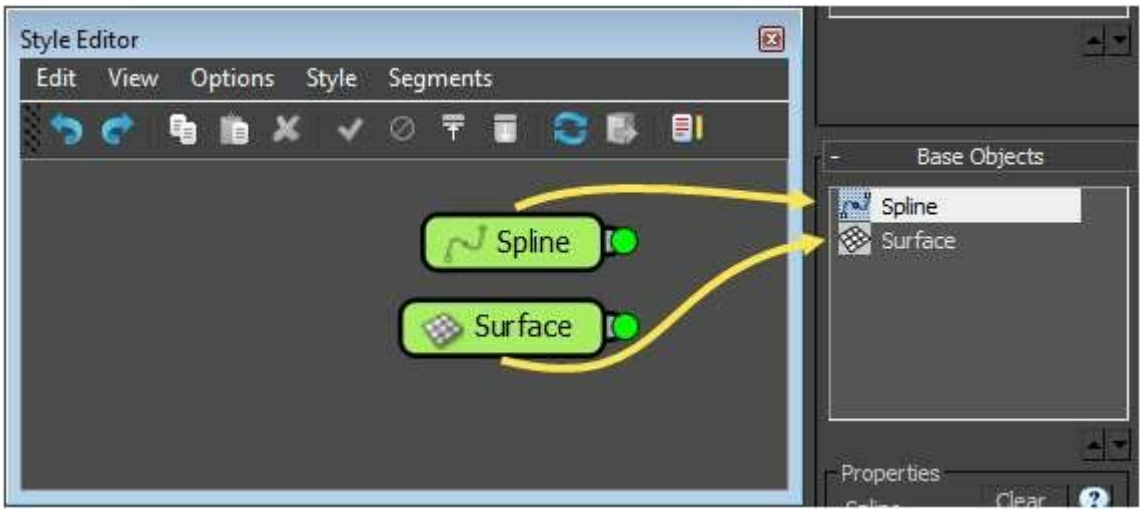
The Parameters rollout lists editable values used by the current style. Unlike the parameters found in the Style rollout, these settings are added by the creator of the rule-set and vary from style to style. In order for a parameter to appear here, there must be an equivalent Numeric node present in the style editor.

1. **Parameters List** Contains the editable parameters used by the current style. To select a parameter, click its name.
2. **Parameter Value** Used to edit the value of the selected parameter.
3. **Parameter Help** Click to display a user created description of the selected parameter.

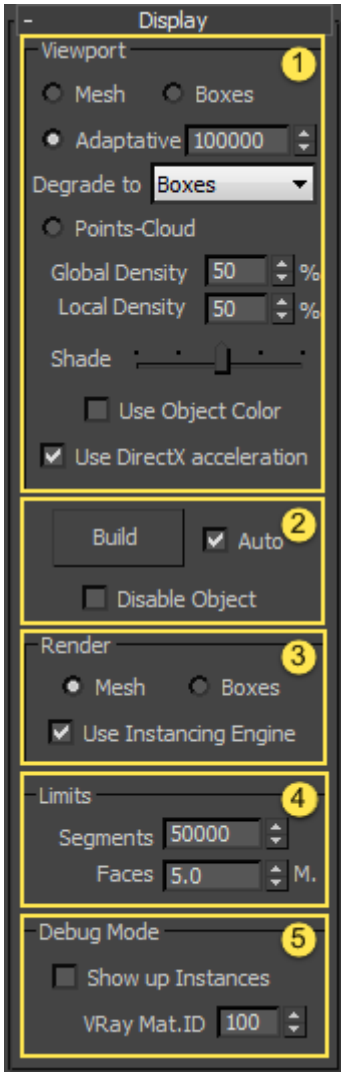


The Base Objects rollout

The Base Objects rollout lists inputs that can be selected from the scene used to generate the current style. For example, a single L1S generator may require a single spline to define the path, a single A2S generator takes two splines, one to define each axis; and both optionally can use a clipping spline and a surface. As with the parameters rollout, in order for a base object to be listed, there will be a corresponding node in the style editor.



1. **Base Objects List** Contains all of the base objects used by the current style. Select a base object from this list to change its properties.
2. **Properties** Used to select splines and surfaces from the scene to be used as base objects. Includes parameters to limit the RailClone object within a set distance along a spline.



The Display rollout

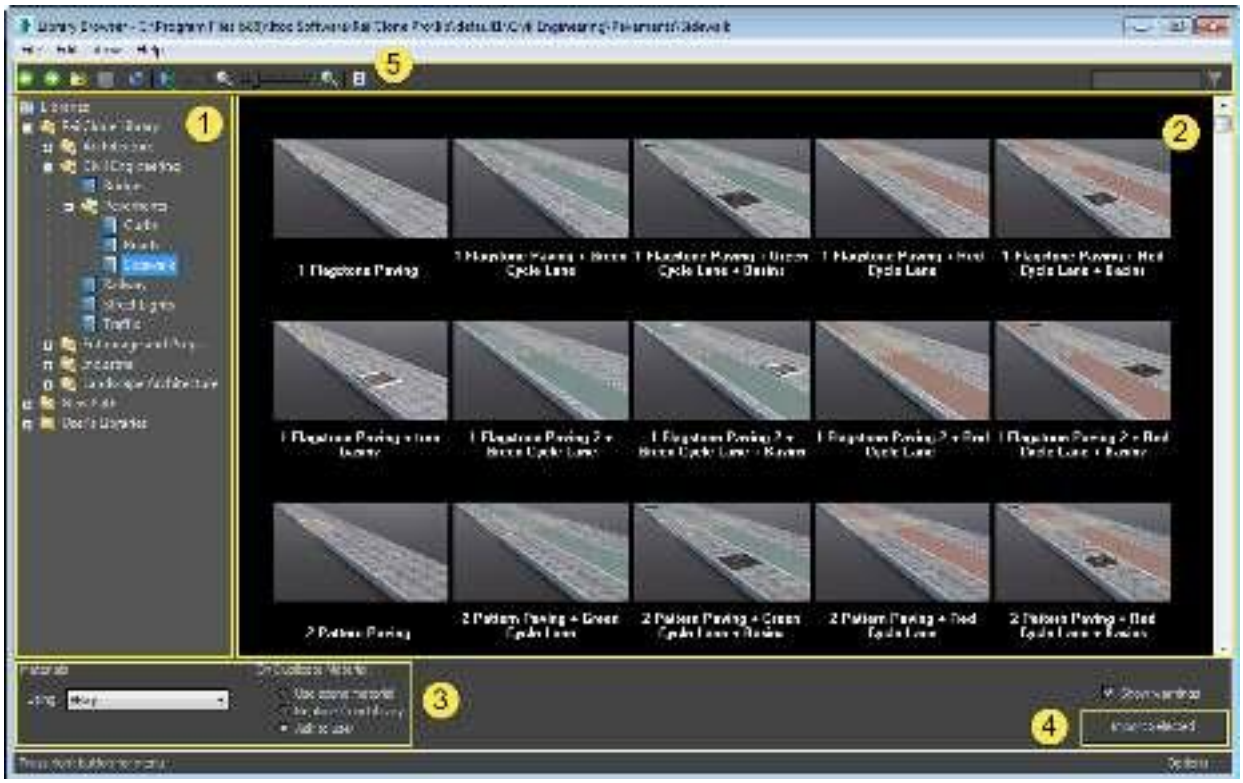
The Display rollout contains options to control the way geometry is previewed in the viewports and generated at render-time.

1. **Viewport** Controls the way in which the RailClone object displays in the viewports. For dense meshes, either box proxies or a points-cloud should be used.
2. **Build** Click to rebuild the RailClone object. Only required if Auto is off.
3. **Render** Used to control the instancing engine.
4. **Limits** Used to set segment and face count thresholds to prevent crashes. If RailClone exceeds these values a warning is displayed and no geometry will be rendered.
5. **Debug Mode (V-Ray Only)** Used to help identify instanced segments. Useful for debugging complex styles.

2 - Library Browser

The Library Browser allows you to load presets that ship with RailClone and easily organise and load your own styles. To open the Library browser, click

in the **Style** rollout (see above). The library Browser's UI is divided into 5 parts:

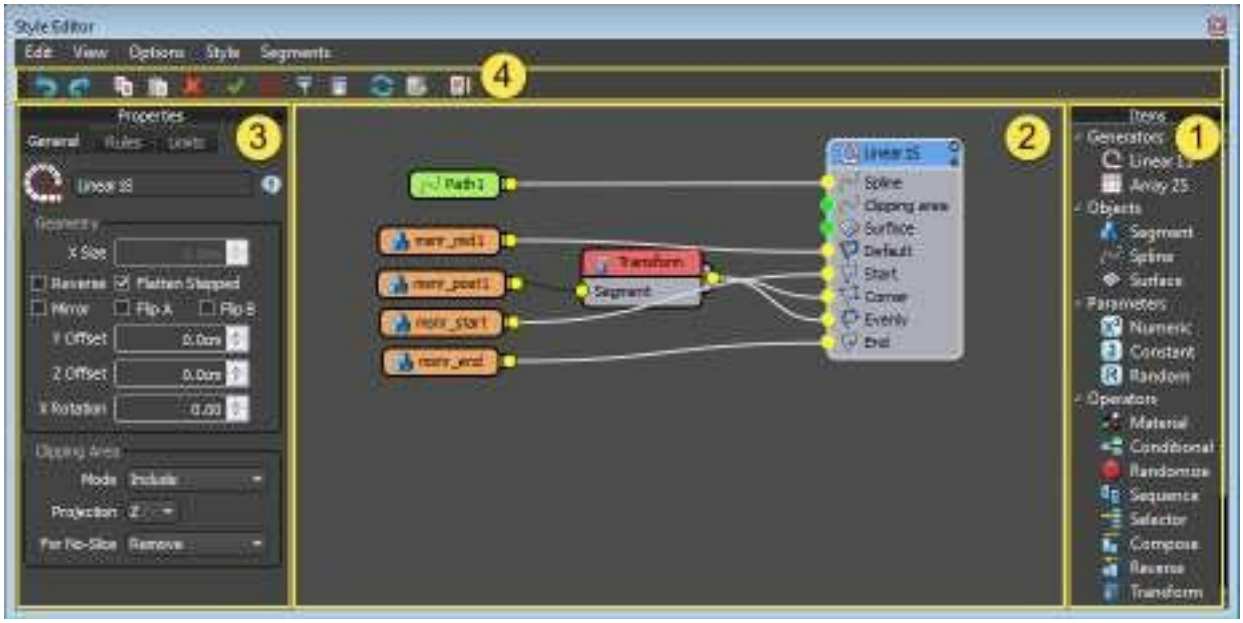


1. **Tree Navigator** Used to locate, edit and select libraries. To navigate, simply use the mouse or keyboard to expand folders and select libraries as you would in Windows Explorer.
2. **Items Grid** Shows the contents of the styles library selected from the Tree Navigator. To select a style, click on a thumbnail image.
3. **Options Bar** Used to select a material library. Can be left on Automatic to detect the current renderer, or you can manually select a material library.
4. **Import Selected** Click this button to load the selected preset.
5. **Toolbar** Provides access to a number of tools to help quickly navigate and modify the library.

3 - Style Editor

The style editor is at the heart of RailClone's workflow. This is the interface used to create the rulesets used to generate procedural geometry. To open the style editor, click on

from the Style rollout (see above).



1. **Items List** Contains node types available for use when designing a new style. To add a new item, simply drag it from the Item List to the Construction View.
2. **Construction View** Used to create Style rule-sets by wiring together Generators, Segments, Operators, and Parameters. Once a node is created in the Construction View, its controls are accessible in the Properties Editor
3. **Properties Editor** Allows you to view and edit the parameters of the selected node. The effects will be updated in real-time in the 3ds Max Viewport.
4. **Tool Bar** Provides buttons for regularly used functions.

Nodes

As you can see from the screenshot above, rule-sets are created by wiring together nodes, and though there are many different types, the interface for all of them is simple and consistent. The parts of a node are as follows:

5.2

1. Icon Indicates the type of node.
2. Input Sockets Used for attaching another node as an input.
3. Name. By default this will be the labelled as the type of node, but it can be changed in the properties editor.
4. Collapse/Expand sockets. Used to hide or unhide the inputs and outputs. Useful to tidy up

complex styles.

5. Output Sockets. Used to Wire a node to other operators or a generator

6. Exported Parameter Exporting parameters allows you to control multiple nodes with a single input, and/or add a Numeric node to enable values to be edited from the modify panel.

7. Exported Attribute Exporting attributes allows you to export the size of a segment's geometry.

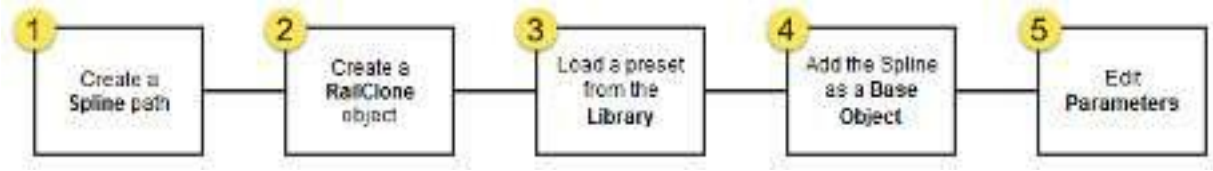
8. Change Order Up/Down Some nodes allow multiple inputs, when that's the case these arrows change the order in which the inputs are processed.

Documentation

For more information please see the Style Editor, General, and Library Browser sections of the online documentation.

6 How to use the library

RailClone includes in excess of 350 predefined styles, allowing you to easily add many common objects to your scenes. The built in presets include roads, curbs, sidewalks, railway lines, bridges, street lights, ducting, trusses, curtain wall systems, theatre seating, railings, scaffold, walls and much much more. Using these styles is simple, often only needing you to create a new RailClone object and a single spline. To explain the process of using a preset, we can break it down into 5 short steps.

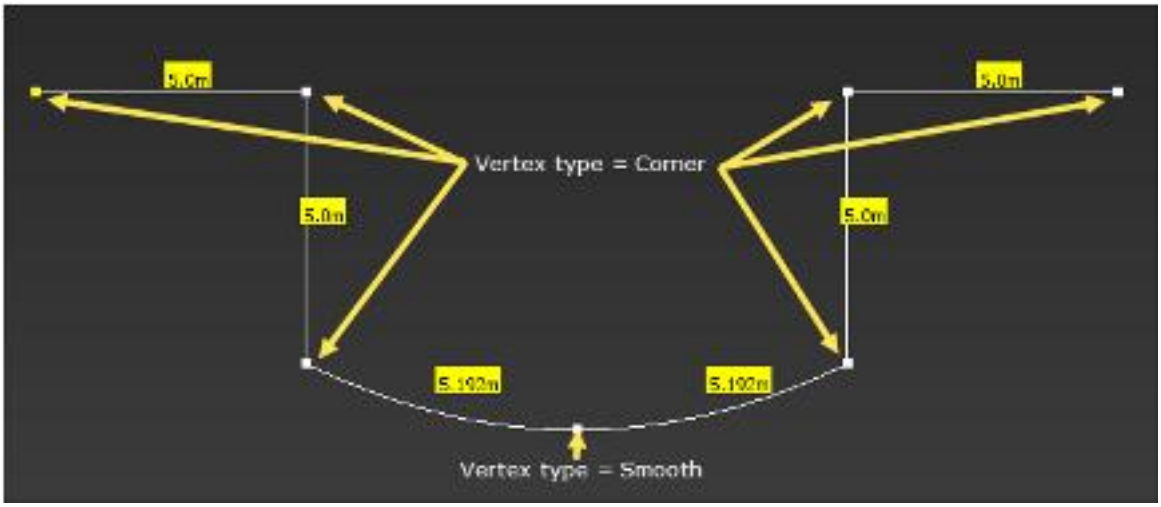


6.1 Exercise: Loading a preset

Step 1 - Create a spline Path

As we saw in Chapter 3, RailClone uses splines to determine not only the size, but the paths for the two array based Generators. Geometry is automatically deformed to follow the path and, In most of the library presets that use them, vertices of the Corner or Bezier-Corner type will generate a corner segment.To illustrate this we'll create a spline:

1. Create a new spline as illustrated in the image below or download and open using_the_library_start.max.



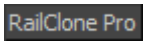
2. Note that the majority of vertex types are set to **Corner** with the exception of the one in the centre of the curved section, This vertex's type is set to **Smooth**.

Step 2 - Create A RailClone object

Next, create a RailClone object just as you would any 3ds Max geometry:

1. Go to 3ds Max's **Command** > **Create** > **Geometry** panel and change the category to **iToo Software.**

2. Click

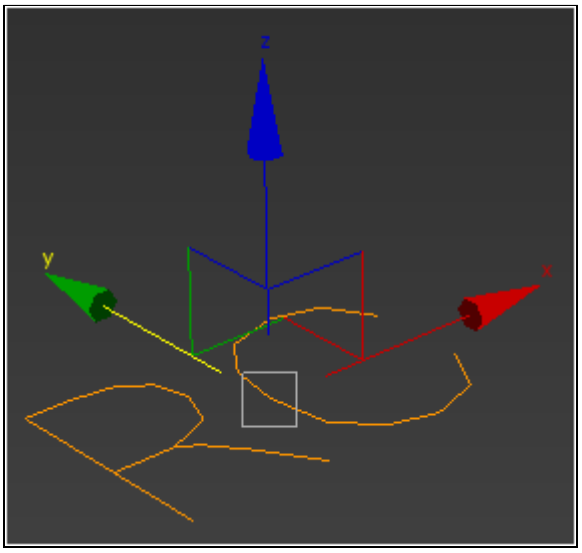


.

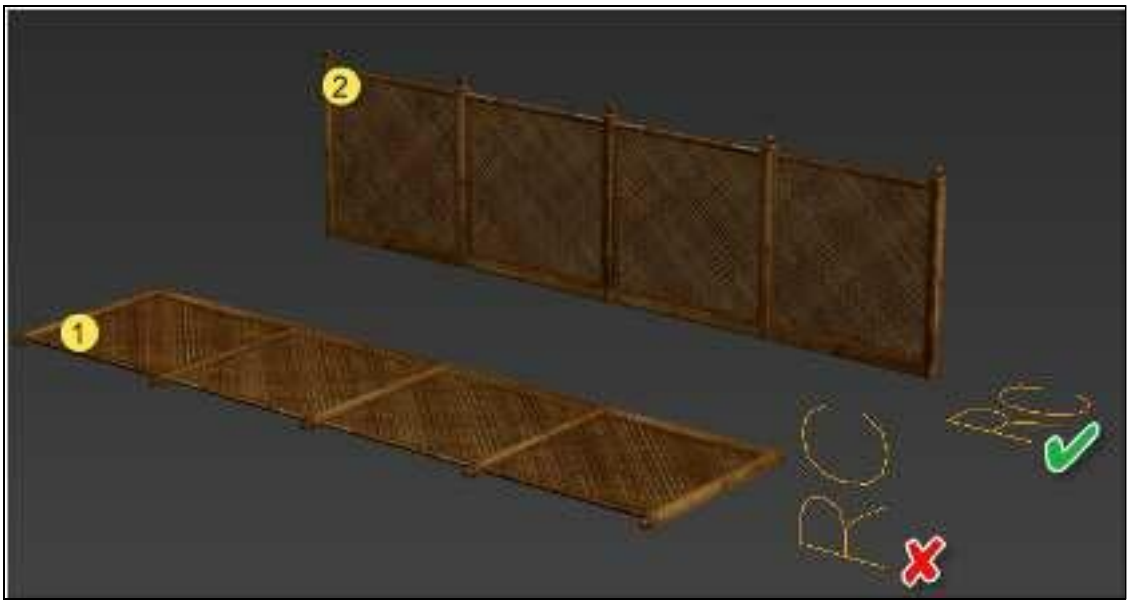
3. Click and drag in the **Top** or **Perspective** viewport to create a RailClone icon.

Why the Top or Perspective Viewport?

RailClone uses its local coordinate system to determine the array's axes, so the orientation of the RC object is important. Here is how the local axes of the RC object looks:



1D Arrays are constructed along the X axis while 2D arrays use the X and Y axes. The axes of the source geometry will always be aligned with the RC objects local axes (we discuss this in more detail in the Intermediate Guide). For example, in most cases you would want a fence's verticals to align with the Z axis, so you should create the RailClone icon in the top or perspective viewport :



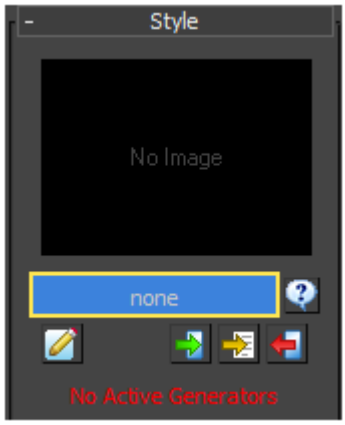
1 - In this example the RC object was created in the left viewport. This is incorrectly oriented for a fence style, though may be correct for other styles such as wall claddings that use a vertical alignment. **2** - In this example the RC object was created in the top viewport, the RC object is oriented correctly for this style.

If you find the alignment is incorrect you can simply rotate the RailClone object to the orientation you require and then reselect the base object(s).

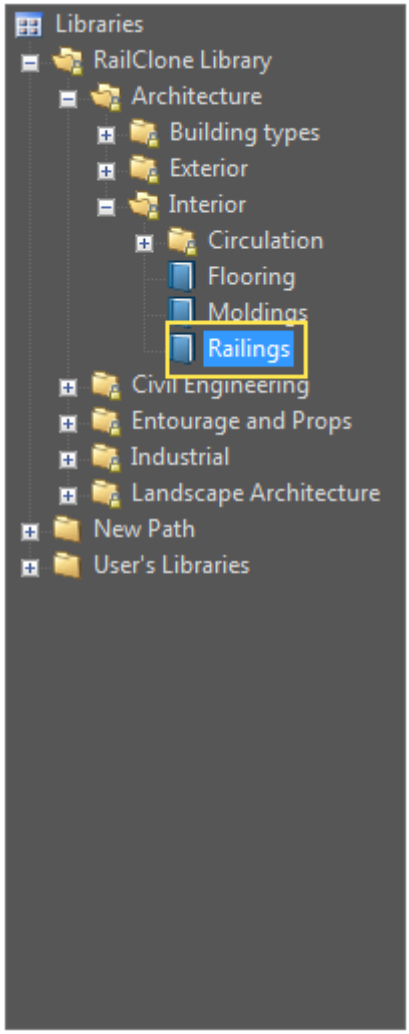
Step 3 - Load a preset from the library

You now have a Base Object and an empty RailClone object. The next step is to load a style. To open the Library Browser and load a preset follow these steps:

1. Make sure the RailClone object is still selected and open the **Modify** panel.
2. Open the **Style** rollout. Click on the **Open RailClone Library Browser** button, this will currently say *"none"*, but will display the name of the style once it has loaded.

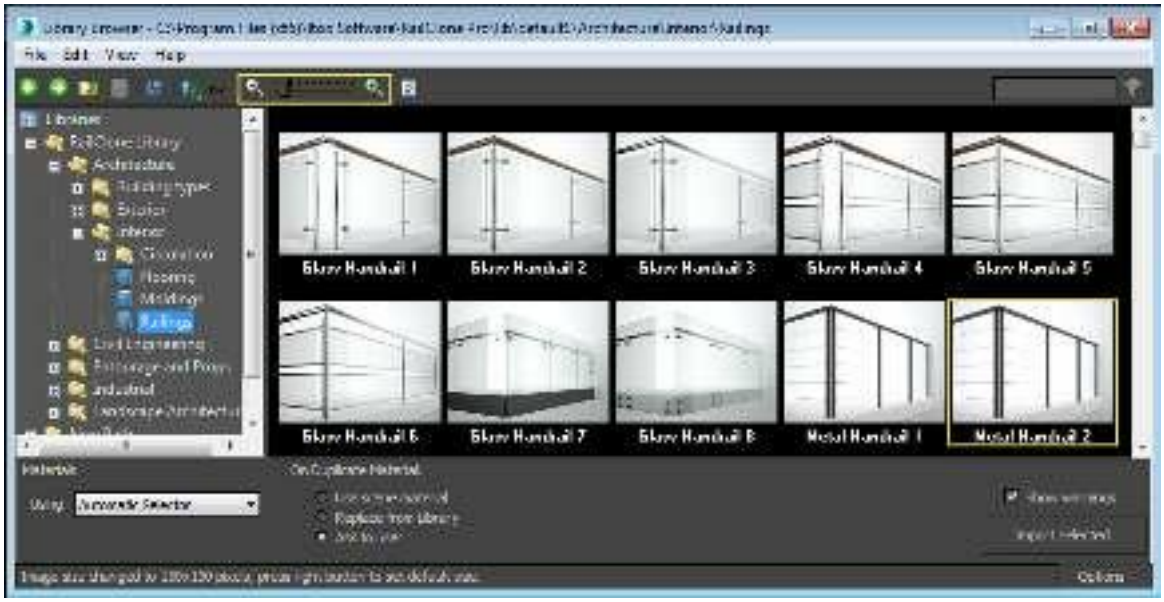


3. The library browser will open. Use the **Tree Navigator** on the left of the window to locate a library. In this example we'll use a **Railings** library which can be found by going to **RailClone Library > Architecture> Interior.**

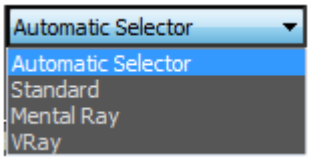


4. When you click the library's name, the items grid will update to display thumbnail previews of the contents. To change the size of the previews you can use the

zoom slider found in the Toolbar. Select **Metal Handrail 2**.



5. Ensure the the **Materials > Using** option is set to **Automatic Selector**. When this is active, RailClone auto-detects the active renderer and merges the most appropriate materials. If you'd prefer to manually specify which materials to merge, use the drop-down menu to select the correct renderer. The built in libraries include materials for Standard, Mental Ray, and V-Ray renderers.

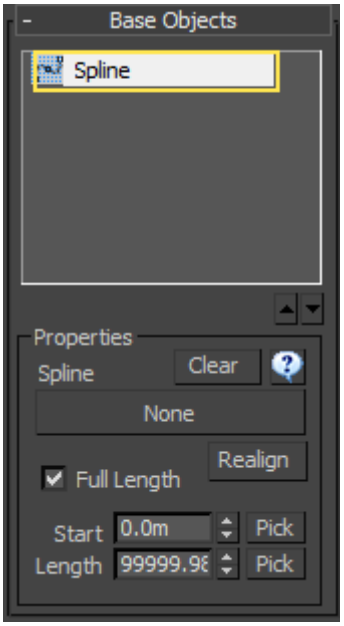


6. Next click **Import Selected**, or double click on the thumbnail. The Library Browser will close and the Style roll-out is updated with the name and preview image of the preset.

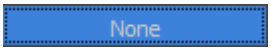
Step 4 - Add the spline as a Base Object

You'll notice there is a red error message displayed in the style rollout. This is because we have loaded the rule-set for this style, but we haven't yet specified the path for it to follow. To do this:

1. Open the **Base Objects** rollout. Here you'll find an option to pick a spline from the scene. (This style only uses 1 spline but others may use more.)

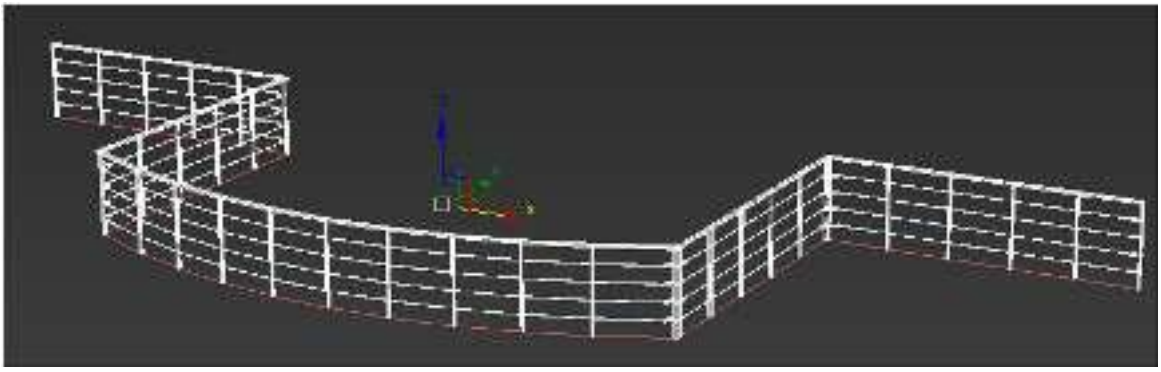


2. Select the **Spline** Base Object and click on



, then select the spline in the scene created in step 1, above.

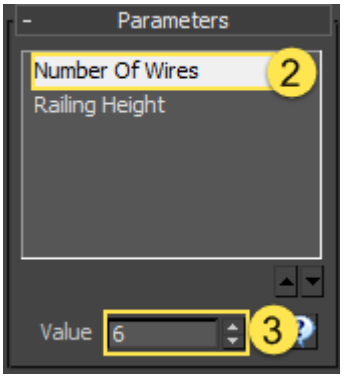
3. The railing style now generates geometry along the spline. Note that a post is added at each vertex except the one in the middle of the curve.



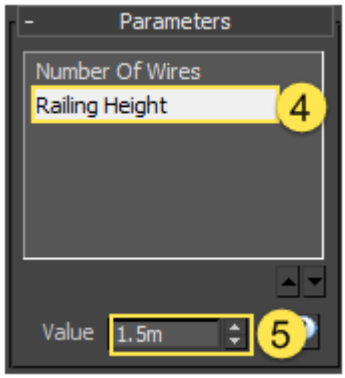
Step 5 - Edit Parameters

Many presets have additional editable parameters that are unique to the style. In this example the height of the railing and the number of wires are adjustable. Follow these steps for an example of how to edit these values:

1. With the RailClone object selected, open the **Parameters** rollout in the Modify panel.
2. Select the parameter called *Number of Wires*.



3. Change the value from *4* to *6*.
4. Select the *Railing Height* parameter.



5. Change the value to *1.5m*.

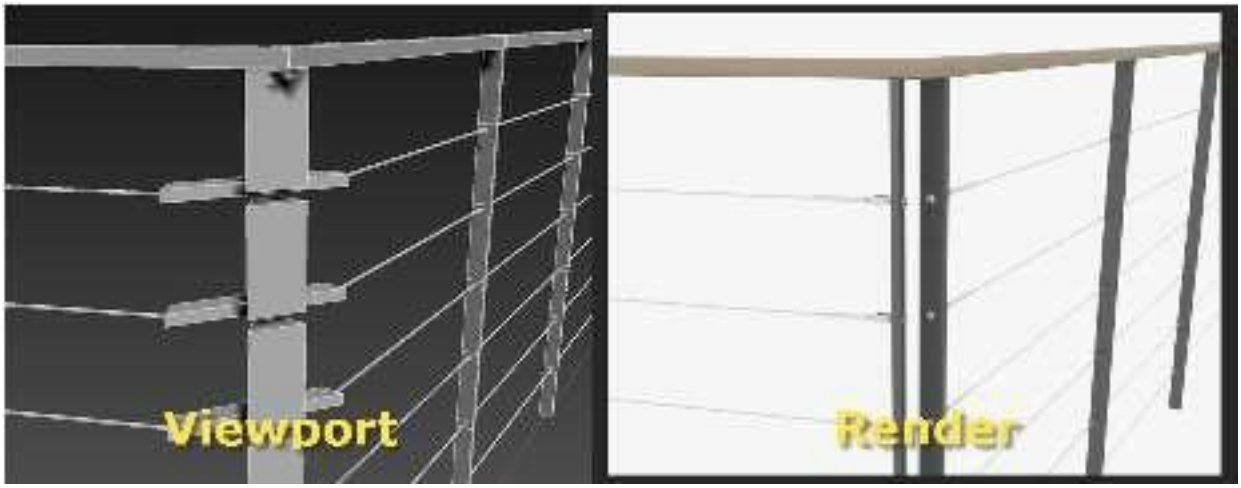
Editing is non-linear

Though we have presented this process as a sequence consisting of 5 steps, one of the great advantages of RailClone is that a style, its parameters, segments, and base objects can be edited at any time and in any order, and the object will update automatically. For example, you can easily edit the spline path, perhaps lengthening a section and adding a new corner, and the geometry is recalculated and updated in real-time. Also, for styles that share the same number of Base Objects, it's possible to replace the existing style with a new preset from the library, and it will be automatically be applied to the currently assigned spine, without needing to reselect it from the scene.

6.2 Exercise: Display Settings

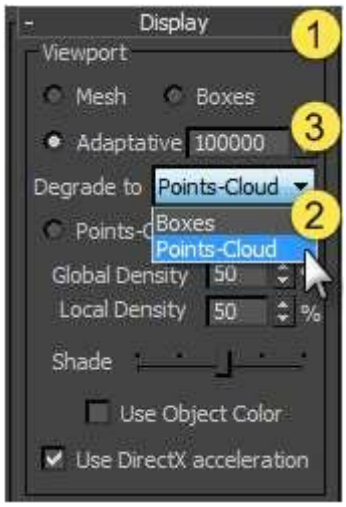
If you look closely at this style in the viewports you'll notice that the geometry does not match the thumbnail preview, in fact It appears to be created using much more primitive geometry. Don't fear! This not an error, but is because the original geometry has been replaced with simplified boxes using one of RailClone's two adaptive display modes. These are designed to keep viewport navigation as fast and interactive as possible, while allowing you to render scenes with huge polygon counts. RailClone automatically switches to this display mode when the face count of the object exceeds a set value, by default 100,000 faces.

If you render the view you'll see that despite the the representation in the viewports, the full geometry is always used at render-time, without any need for user-intervention.



The second adaptive display mode allows you to display the geometry as a point-cloud. This mode is useful if you need a more accurate representation of the geometry in the viewports, and especially useful while you're designing complex high-poly styles. To switch on Points-Cloud display mode, follow these steps:

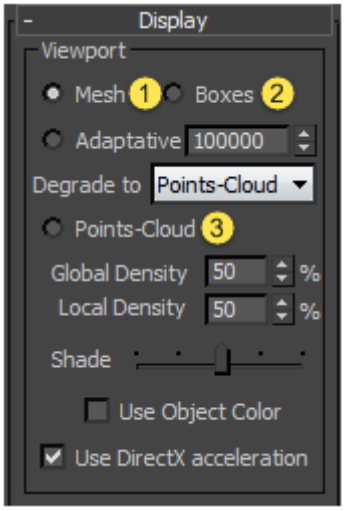
1. With the RailClone object still selected, open the **Display** rollout.



2. Click on the **Degrade to** drop-down list, and select **Points-Cloud**.
3. From this menu you can also change the face threshold value that the object must exceed in order to trigger a simplified display mode. To do this enter a new value on the **Adaptive** value box.

The railing will now be displayed in the viewport as a points-cloud, but if you render you will see that the full geometry is used at render-time, again without any need for user-intervention.

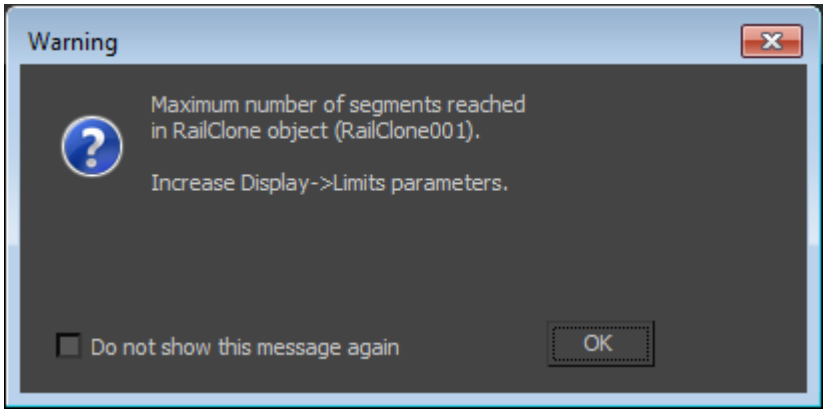
It is also possible to set the display mode manually, irrespective of the number of faces. To do this, open the Display rollout and select one of these 3 options:



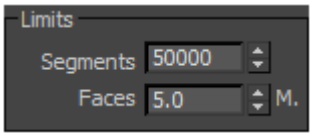
1. **Mesh** Turns off the preview modes and displays the full mesh. Often useful when first creating a style but be careful using this mode on large objects as huge poly-counts can easily be created that slow down performance.
2. **Boxes** Forces RailClone to display geometry as boxes.
3. **Points-Cloud** Forces RailClone to display geometry as a points-cloud.

6.3 What to do if RailClone exceeds its limits.

Occasionally, when creating a very large RailClone object, you may get a warning telling you that the maximum number of segments has been reached.

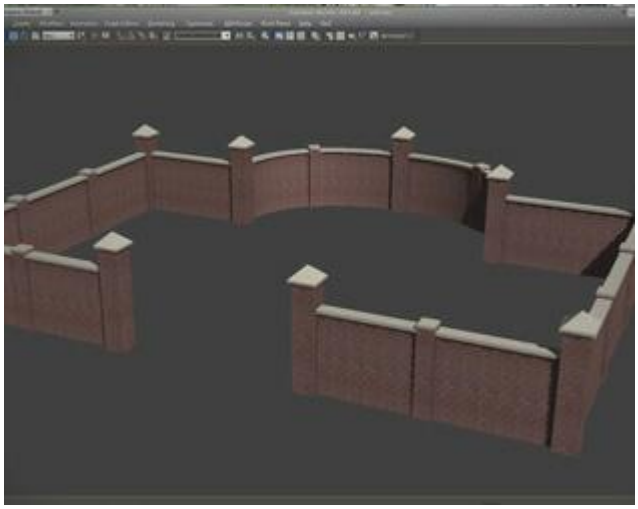


This is because RailClone imposes a segment and face count limit to prevent users from accidentally creating RC objects so large they risk crashing 3ds Max. This limit can be increased by going to the display rollout and adjusting the **Segments** and **Faces** values.



When calculating the face count, instanced geometry is not taken into consideration.

6.4 Related Tutorials:



building_a_masonry_wall

Masonry Wall Tutorial

The first part of this tutorial includes a section on loading styles from the library browser that may provide a useful recap for this chapter.

create_a_seaside_promenade

Seaside Promenade

Part one of this tutorial explains how to load a metal railing preset from the library and apply it to a spline Base Object.

Documentation

For more information please see the Customizing the Library, Display Settings, Parameters, Path Base Objects, and Surface Base Objects sections of the online documentation.

7 How to edit a RailClone style

The presets in the library can be easily modified and adapted to create your own styles. By adjusting the rule-sets and editing or even replacing the source geometry, the library presets are a springboard for creating your own styles and gaining a deeper understanding of how RailClone works.

7.1 Exercise: Using the Library to Learn RailClone

Reverse engineering presets is a great way to understand how to construct new rule-sets and create source geometry for RailClone. We strongly recommend you take some time to examining the node-trees for a selection of presets to demystify how they work.

For example, let's look at a deceptively complex looking preset from the Stadium library:

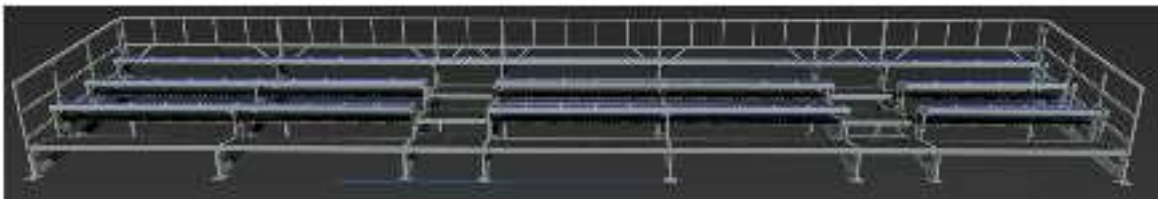
1. Start a new scene and draw a spline approximately 12m in length.
2. Create a new RailClone object, select and load the **Architecture** > **Building Types** > **Stadium** > **Bleachers** > **Metallic Bleachers 3SR** preset from the library. (

   PRO only

   Lite users can select **Metallic Bleachers 3**. The style uses fewer segments but many of the same principles will apply)

Top: Metallic Bleachers 3SR (Pro only) | Bottom: Metallic Bleachers 3 (Lite Users)

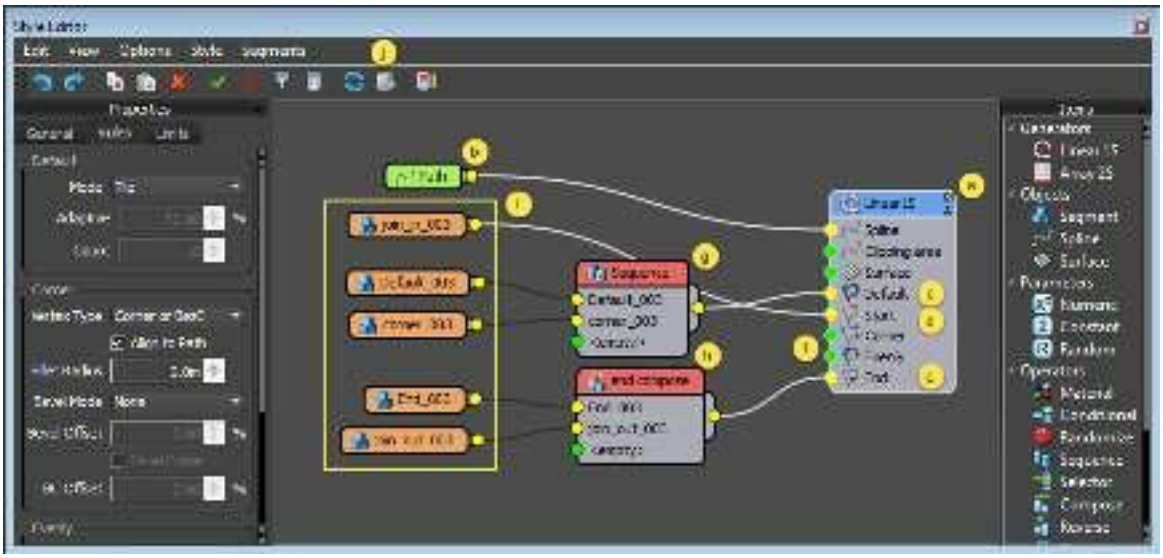
1. From the **Base Objects** rollout, click on **Spline** and select the path created in step 1. You'll now see a bleacher as shown below.



2. At first glance this looks like a complex style. To look at how it works in detail, you can examine the rule-set. To do this, go the the **Style** rollout and click

to open the **Style Editor**.

3. The style editor will open showing the style's node-tree. You can see that actually this rule-set is quite straightforward.



By examining this construction, we can learn a few things about this style and the way that RailClone uses these nodes to create a rule-set.

a. The preset uses a single **Linear Generator**. This generator creates a one dimensional array using a length measurement or a spline as a path.

b. The preset uses a spline to determine the **Path** for the 1D Array. To use a spline you must wire a **Path** object to the generator's **Spline** input.

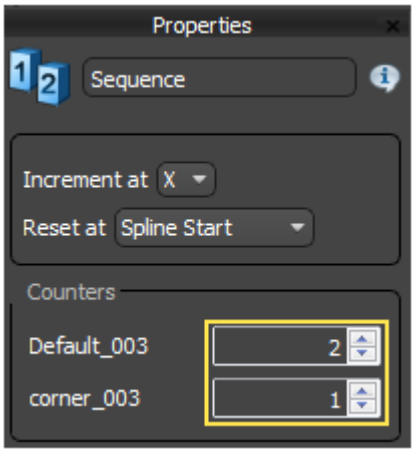
c. The preset has segments attached to the **Default** input. As the name suggests, this geometry is used by default unless the rule-set specifies that geometry from one of the other inputs should be used instead (as we'll see next).

d. The preset has a segment attached to the **Start** input. Geometry in the start input will be used **once**, at the **beginning** of the spline.

e. The preset has segments attached to the **End** input. Geometry in the end input will be used **once**, at the **end** of the spline.

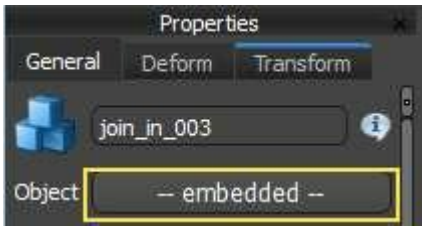
f. The preset does not use the Corner or Evenly inputs. This preset is designed to work on straight or curved splines, not splines with hard turns.

g. The preset uses a **Sequence** operator in the **Default** input. Operators allow you to manipulate segments, we'll discuss them in more detail in chapter 11. But by looking at the style you can deduce that the sequence operator used here is creating a repeating pattern in the array by repeating the segment in input 1 twice, followed by the segment in Input 2. To see how many times a segment is repeated by a sequence operator you click on the node and examine the **Count** values found in the **PropertiesEditor.**



h. The preset uses a **Compose** operator in the **End** input. Compose operators allow you to combine segments, this is useful for adding multiple segments to an input ordinarily designed for just one, such as Start, End, Corner or Evenly.

i. The style uses 5 **Segments**. Each segment node represents a piece of geometry added to and duplicated in the array. To fully understand how a rule-set builds a RC object, it's helpful to be able to see the pieces of geometry from which it is constructed. In library styles the geometry is no longer in the scene but stored inside the RailClone object itself. You can see this by clicking on a Segment node and going to the **Properties**, if the geometry is stored inside the RC object **Embedded** will be displayed in place of the object's name.



In the next sections we'll look at how to get that geometry back into the scene so it can be examined, edited and replaced.

Extracting Geometry

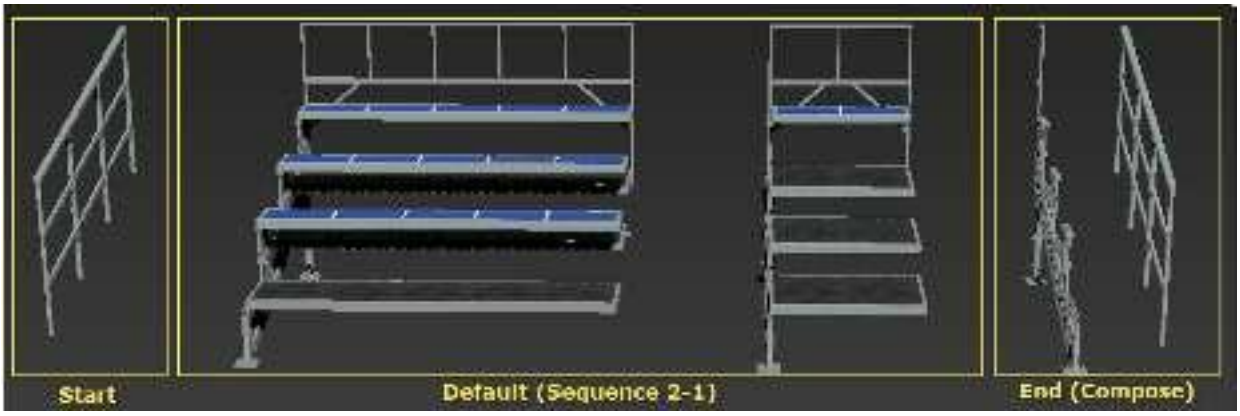
Once geometry is assigned to a Segment it is possible to delete the original object and , as we've seen, the RailClone object stores a copy of the mesh internally. However, to edit geometry and adapt styles it may be necessary to extract this object back into the scene. To extract the geometry of the Bleachers style follow these steps.

1. In the Style editor, select one of the 5 Segment nodes.
2. Click

   from the tool bar or go to **Segments>Extract** from the menu.
3. The Segment's source object will be created in the scene at its original coordinates.
4. Do the same for the remaining 4 segments.

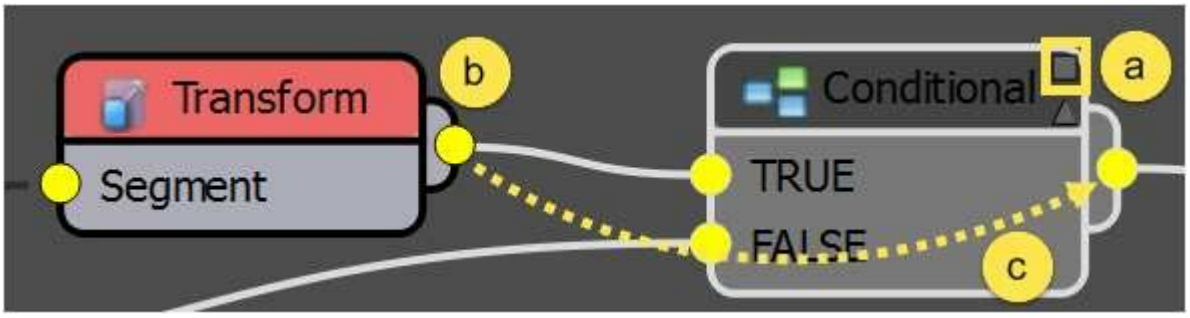
You can now see the 5 segments that make up the style. **Understanding how the geometry and the rule-set works together is key to mastering the RailClone workflow.** In this example you can see that a railing starts the array. The default part of the array, between the start and then end, is made using a Sequence operator to create a repeating pattern of seating and steps (note the supporting legs on the left hand side of both segments). To finish the array we repeat the railing used at the start but before that we need another set of legs, these two components are combined using a Compose operator.



Working with RailClone is a little like creating the pieces of a jigsaw and then providing detailed instructions on how they all fit back together. Let's try another example.

Exercise: Disable Nodes to deduce their Function

When analysing library styles, one of the most useful ways to work out what a node is contributing to the rule-set is to toggle it on and off by clicking the the small cross in the top right hand corner. This will disable the function of that node and in most cases the first input will pass through. We'll use this tip to work out how to construct a balance shelving style by reverse engineering the preset.



In this illustration the Conditional Node (a) has been disabled. The transform node (b) passes through, ignoring the Conditional node as illustrated by the dotted arrow (c)

1. Start a new scene and draw a spline approximately 1m in length.
2. Create a new RailClone object, select and load the **Entourage and Props > Shelving and Storage > Balance Shelving 1** preset from the library.

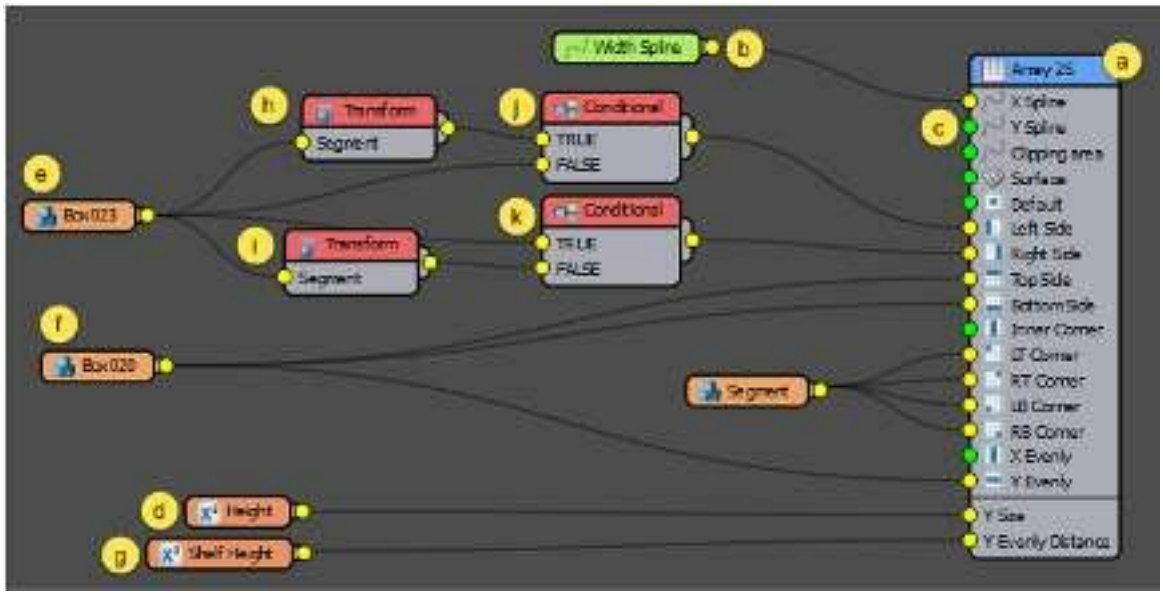
3. From the **Base Objects** rollout, click on **Spline** and select the path created in step 1. You'll now see shelving as shown below.

If you open the Parameters rollout you'll see two values you can edit to change the overall height, and the spacing between shelves.

4. Go the the **Style** rollout and click

to open the **Style Editor** and examine how this preset is made.

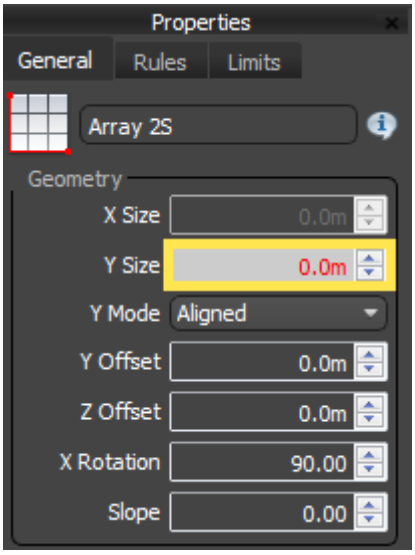
5. As in the previous example there are a few things you can learn just by examining the nodes:



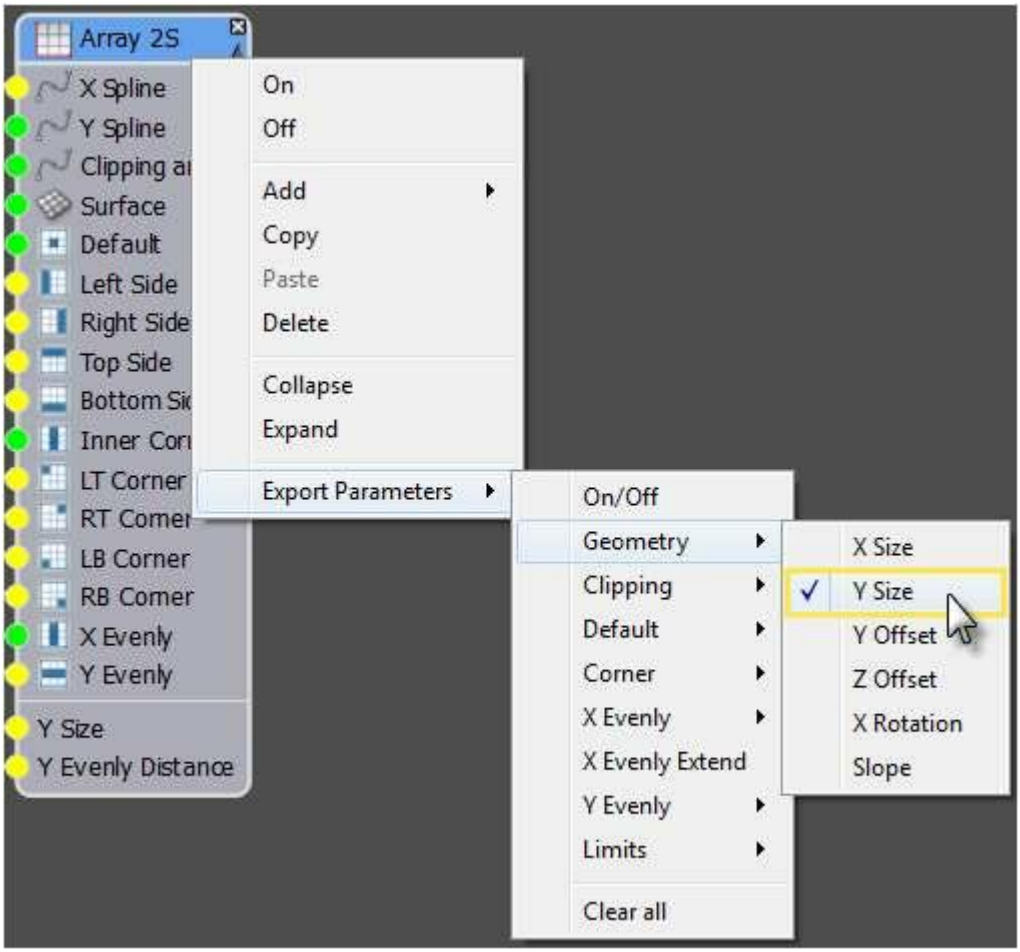
   a. The preset uses a single **A2S Generator**. This generator creates a two dimensional array to create a shelf unit that has an adjustable height and width.

   b. The preset uses a spline to determine the **Path** for the **X** axis of the array. To select a spline you must wire **Path** objects to the generator's **Spline** inputs.

   c. The preset **does not** use a spline to determine the path for the **Y** axis. This means that the height must be set using the generator's **Y Size** property. You can find this by going to the Properties Panel. Note that the value is displayed in red, this means that this property has been **Exported**.



d. The preset has an **Exported Y Size** property wired to a **Numeric** node. This allows you to adjust the distance between shelves from the **Parameters** rollout in the Modify panel. To export a parameter you right click on the node and pick a property from the **Export Parameters** menu.



We'll explore this feature in more detail in *Next Steps with RailClone: A guide for intermediate users* .

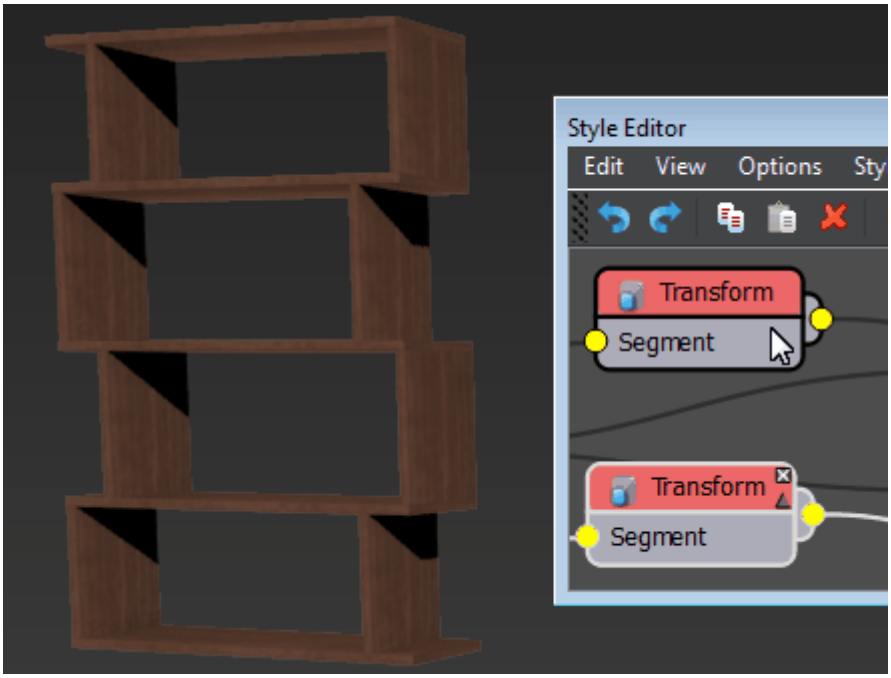
To work out what the remaining nodes do, try toggling them on and off to observe the results:

e. Turning off the Segment node named *Box 23* results in the sides of shelf disappearing. This is logical as the segment is attached to the **Left** and **Right** inputs.

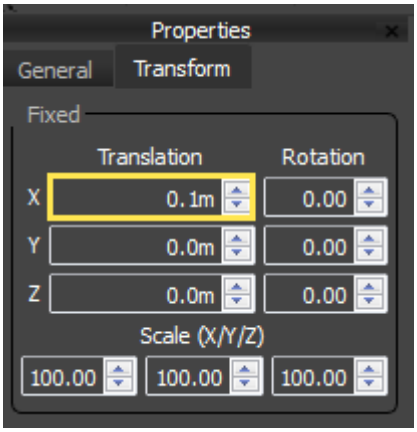
f. Turning off the Segment node named Box020 results in all the horizontal elements disappearing. Again this is logical, this segment is wired to the **Top**, **Bottom** and **Y Evenly** inputs.

g. The preset has an **Exported Y Evenly** property wired to a **Numeric** node. This allows you to change the shelf spacing from the Parameters rollout in the Modify panel.

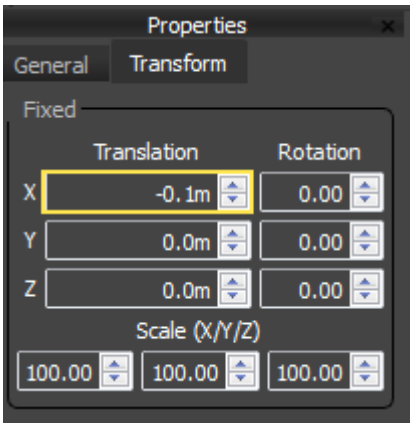
h. Take a look at the two **Transform** operators. Turning off the topmost Transform operator straightens the left side of the shelf.



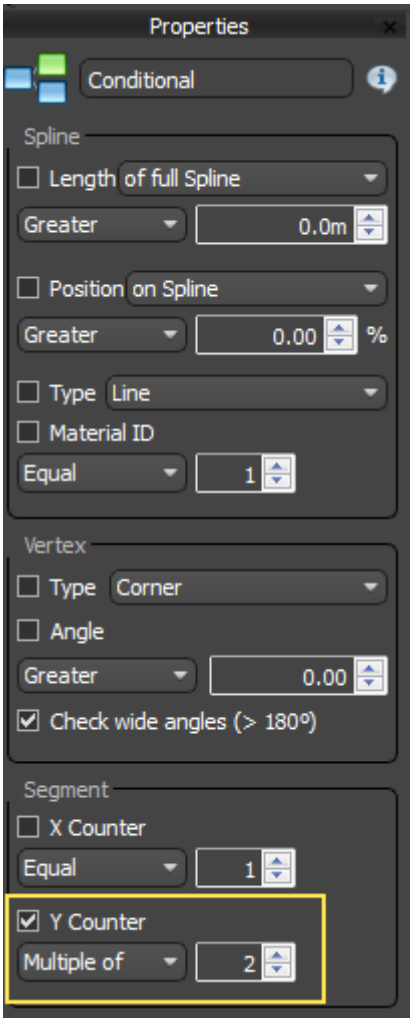
You can therefore deduce that it is this operator is offsetting the segment. If you look at the Transform Operator's **Properties** > **Transform** > **Fixed** >**Translation** > **X** value, you can see that it is indeed being offset by 0.1m.



i. Turning off the bottom **Transform** operator straighten the right side.The transform operator is also offsetting the segment, but this time using a negative value.



j. + k. The two transform operators are wired to **Conditional** nodes. If you look closely at the wiring you'll notice that the Segment node is wired directly to the **True** input of the first Conditional node and the **False** input of the second, the Transform nodes are wired to **False** input of the first Conditional node and the **True** input of the second. When you turn one of the Conditional nodes off, only the segment wired to the True input is used. To check out how the conditional operator works, open the properties panel and look for the activated check-box.



The Y Counter check box is ticked. By examining the settings, You can see that this node returns the Segment wired to the True input when the Segment counter for the Y Axis is a multiple of 2. This creates the alternating pattern on the sides of the shelf unit.

By systematically going through the nodes and toggling them on and off you can get a pretty good understanding of how this rule-set works. In this section you've only examined two presets, but if you're new to RailClone, it is often helpful to analyse something similar from the library before creating a new style.

7.2 Exercise: Editing Library Presets

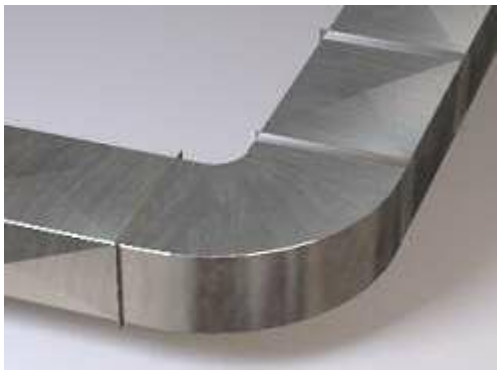
Sometimes you want to use a preset from the library but it isn't quite right for the project, fortunately styles loaded from the library are fully editable. It may be that you just need to adjust a few settings, or you could use a preset as the basis for a completely new style. The following section examines some of the elements of a preset that are most frequently and usefully edited to adapt the style.

Adjust commonly used Generator Parameters.

Rotate

The parameter rotates the geometry around the X Spline. To use it:

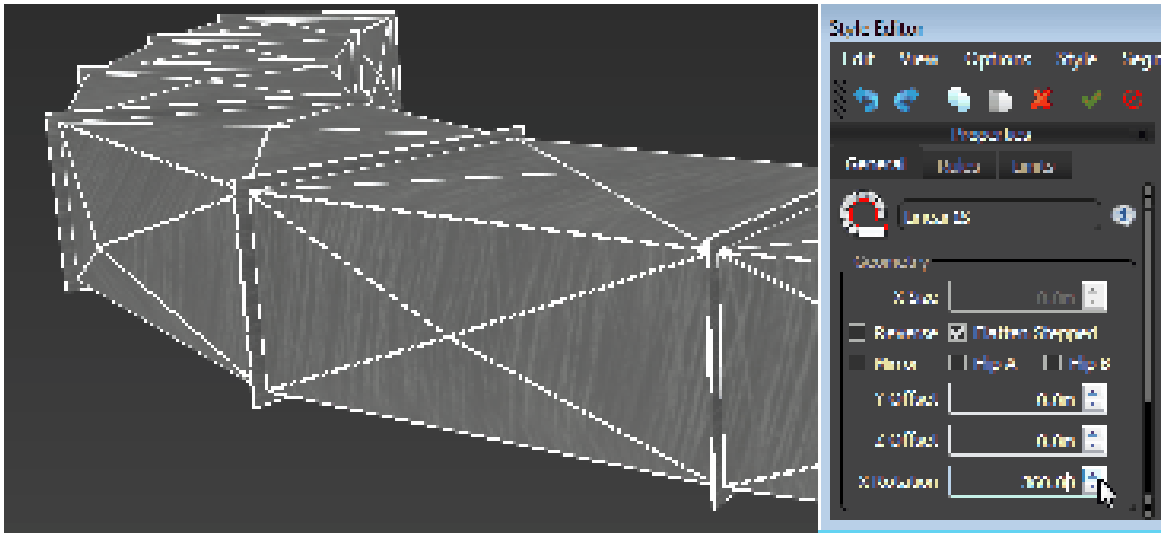
1. Create a new RailClone object, select and load the **Industrial** > **Ducting** > **Rectangular Ducting** preset from the library and add a spline as the Base Object.



2. Go the the **Style** rollout and click

to open the **Style Editor**.

3. Select the Generator and edit the **Properties** > **Geometry** > **X Rotation** value. Editing this value rotates the whole style around the spline.



Reverse

This parameter reverses the direction of the path. This has the same effect as applying a "Reverse" command to an editable spline. To use it:

1. Create a new RailClone object, select and load the **Civil Engineering** > **Street Lights** > **Street Light 3 Single** preset from the library and add a spline as the Base Object.

2. Go the the **Style** rollout and click

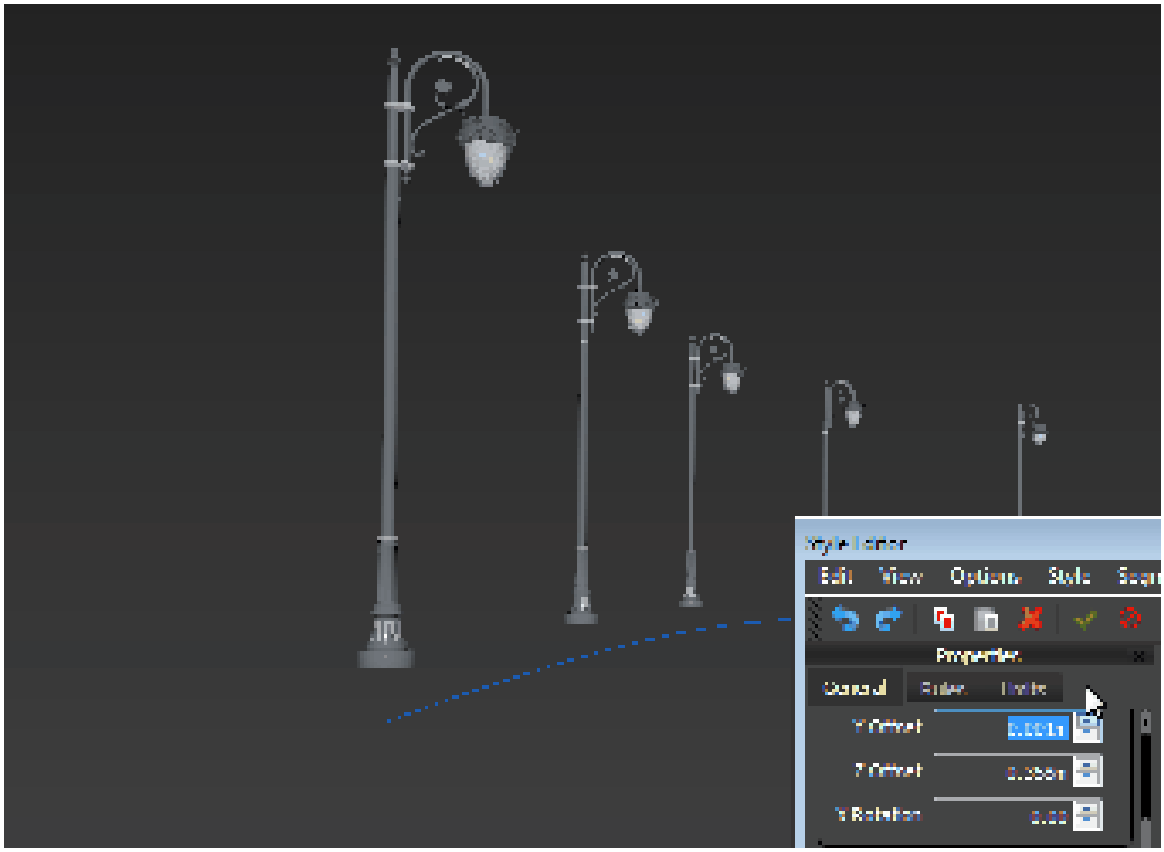
to open the **Style Editor**.

3. Select the Generator and toggle the **Reverse** check-box. The lights will flip to face the other side of the spline.

Offset Y/Z

Y Offset applies an offset perpendicular to the path's direction, **Z Offset** adjusts the position of the geometry on the RailClone Objects local Z axis. To use these settings:

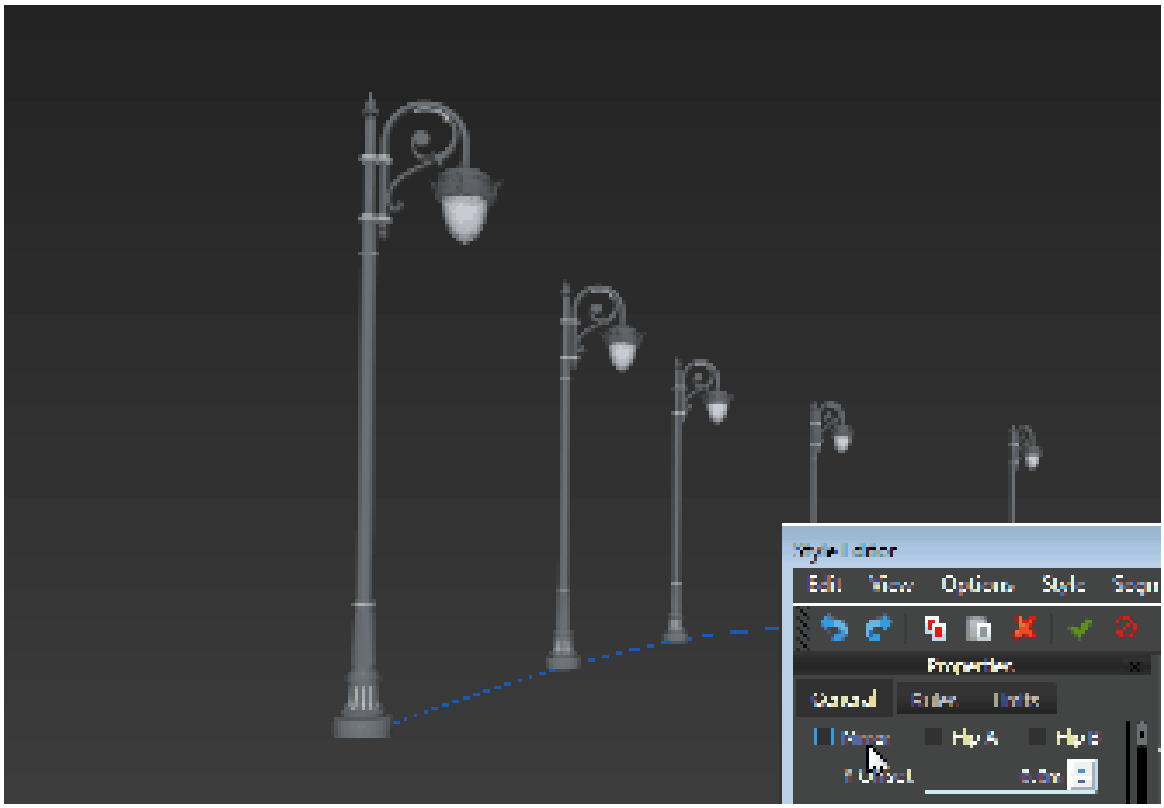
1. Use the Street Light preset from the previous example and open the Generator's properties.
2. Edit the **Y Offset** value, the lights will offset perpendicular to the path.
3. Edit the **Z Offset** value, the lights will move up and down.



Mirror, Flip A, and Flip B

These parameters clone the RailClone geometry using the path as a mirror axis. This is particularly useful for presets such as sidewalks and street-lights where you might typically have a spline that defines the centre of the road and you require these styles on both sides. The Flip options allow you to flip the geometry where A is the original array and B is the copy created with the Mirror feature.To use this feature:

1. Continue with the Street Light preset from the previous example and open the Generator's properties.
2. Click **Mirror**.
3. Increase the **Y Offset** value. you will now have two parallel arrays. At the moment all the lights are facing in the same direction.
4. Click **Flip A** to switch the lights on one side of the array so that both sets face towards one another.

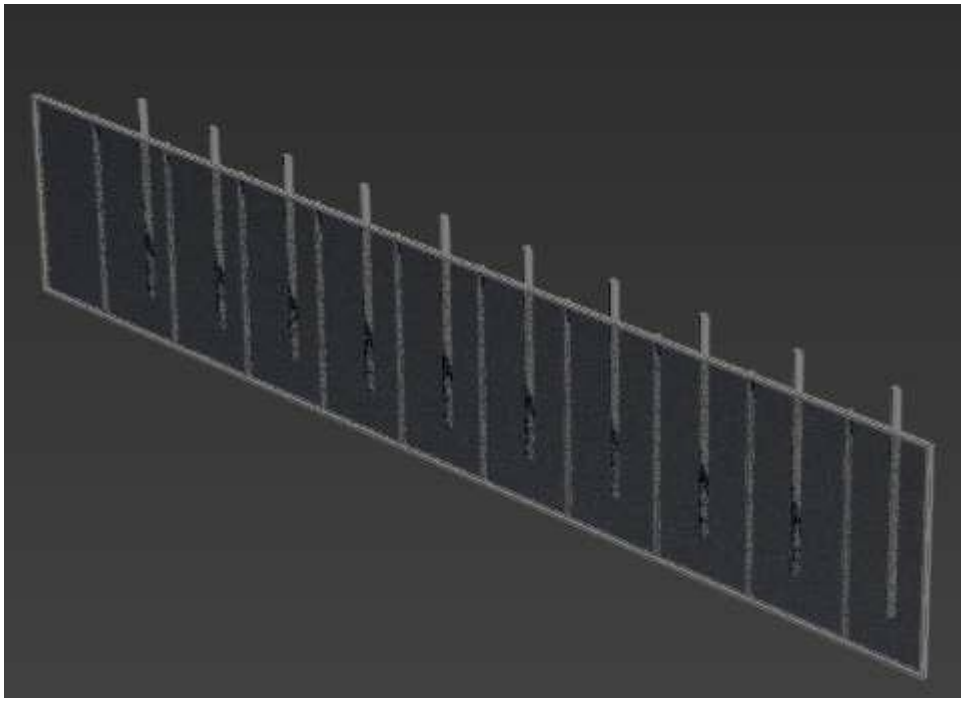


Evenly Distance

Evenly spaced segments are placed at regular intervals along the spline. In some presets, the distance value between evenly segments has been exported and can be adjusted from the Parameters rollout, for styles missing this parameter you can adjust the spacing by following the steps outlined in the following example.

1. Download and open *exercise_curtain_wall_adapt.max*. This file contains the objects needed for the next 3 exercises.
2. Create a new RailClone object, select and load the **Architecture** > **Exterior** > **Facade** > **Curtain Wall** > **Curtain Wall Curve 01** preset from the library.
   PRO only

   
3. From the **Base Objects** rollout, click on **Spline** and select the spline called *Line001* from the scene . You'll now see a façade as shown below.



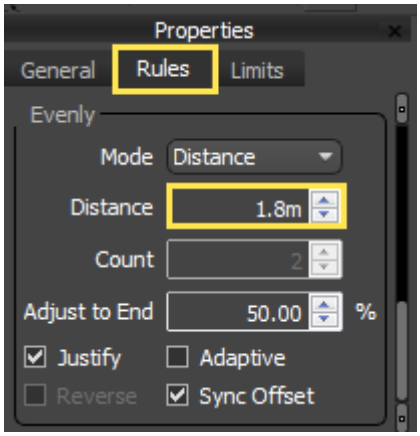
4. Go the the **Style** rollout and click

   to open the **Style Editor**.

5. Go to the Generator's **Properties** and switch to the **Rules** tab.

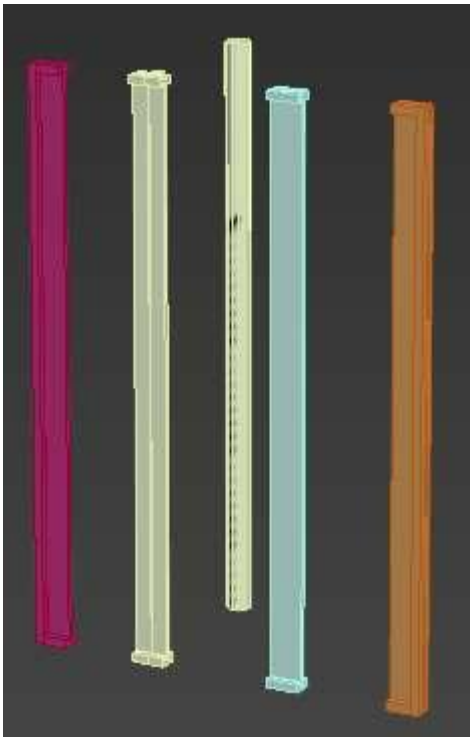
6. At the bottom of this tab you'll see the **Evenly** group, adjust the **Distance** parameter to control the spacing between Evenly Segments.



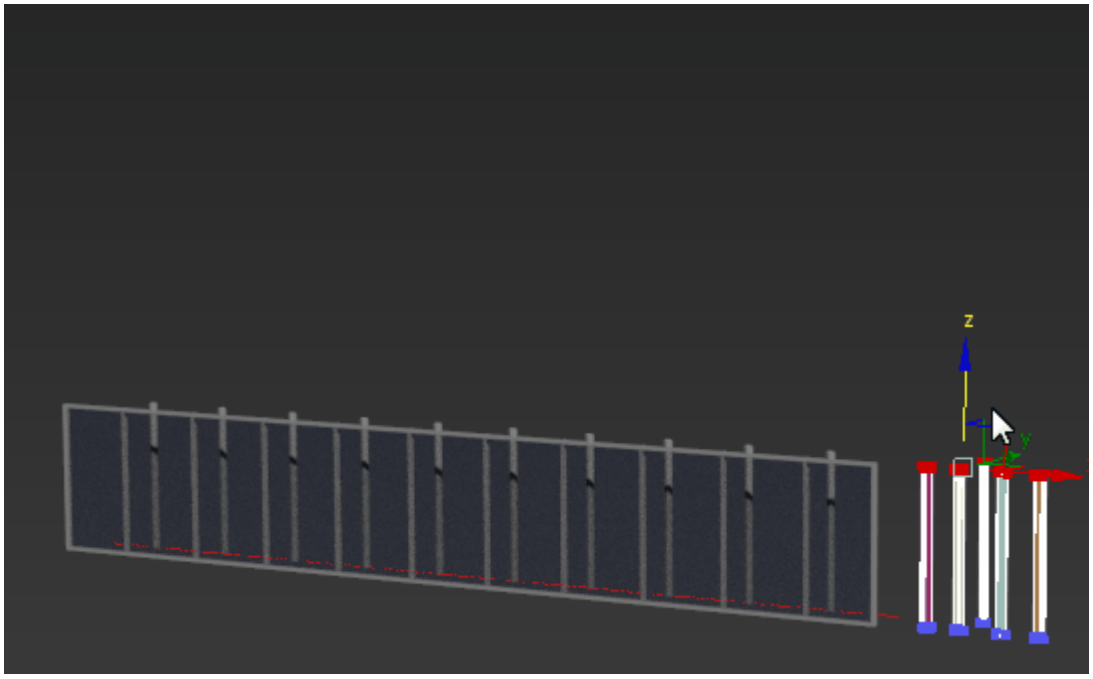
Extract and edit Geometry

RailClone segment's geometry is embedded in the object in a similar fashion to Max's compound objects. However as we've seen, these can be extracted allowing you to easily adapt existing styles. In this example we will edit the height of the geometry used in the curtain wall style from the previous example.

1. Open the Style Editor and select the **Segment** node wired to the Generator's **Default** input. Click  to extract the segment.
2. Do the same for the remaining 3 Segments. You will now have these 5 mesh objects in the scene.



3. Let's imagine that In the current style the wall is not tall enough and we need it to be larger. Once extracted, the geometry is instanced to the RailClone object, if you edit the extracted mesh the RC object will also update automatically.
4. Select all 5 extracted wall meshes and add an **Edit Poly** modifier. Select the vertices at the top of the mesh and use the move tool to increase the height. As you can see, the style updates live in the viewports. Once you're happy with the changes , the geometry can be safely deleted and RailClone will continue to retain an updated copy internally.



Swap Geometry

Once you understand how a style works you can use it as a template and add your own geometry. In this example we'll keep the same façade preset, but swap the geometry to create a completely different style. To do this we'll use the four meshes included in *exercise_curtain_wall_adapt.max*.

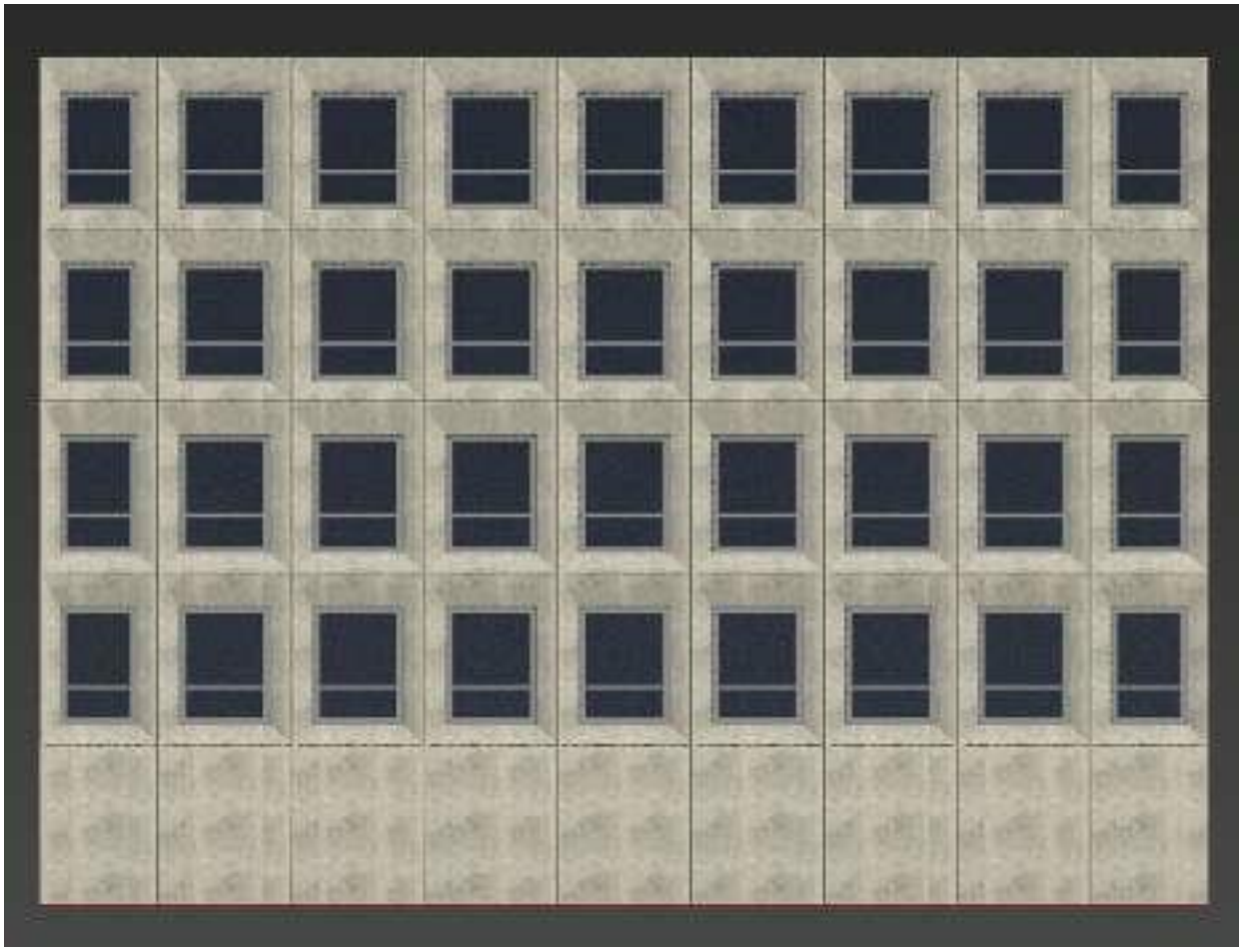
1. Keep the style from the previous example and open the **Style Editor**.
2. To swap geometry, select the Segment wired to the **Default** Input, go to the **Properties** panel and click the **Pick Object** button (

), then select the mesh in the scene called ***Default***.

3. Do the same for the other Segments. Select the segment wired to the **Start** input, go to the **Properties** panel and click the **Pick Object** button , then select the object in the scene called ***Start***.
4. Select the segment wired to the **End** input, go to the **Properties** panel, click the **Pick Objec** t button, and select the object in the scene called ***End***.
5. Finally, select the segment wired to the **Evenly** input, go to the **Properties** panel, click the **Pick Object** button, and select the object in the scene called ***Evenly***.

By following this procedure, what started as a stock preset will have changed from this...

... to this...



...just by swapping the geometry.

Starting with library styles either as a learning aid or as a template for new geometry is a powerful way to get started with RailClone. In the next chapter we'll dive into the L1S generator and create a new style from scratch.

7.3 Related Tutorials

stadium

Stadium Tutorial Part 1

In the first part of this tutorial we learn how to use some of the default library styles that are included with RailClone, looking at how they can be used out-of-the-box, or adapted to solve individual problems. In this tutorial you create the metal railings, concrete barriers and glass façades that run around the entire stadium

civil_view_interop.

Using RailClone with Civil View

In this tutorial we explore how to apply presets from the RailClone library to Base Objects imported from Civil 3D. Many of the styles are edited using Mirror, X/Y Offset, and Flip A/B.

Documentation

For more information please see the Customizing the Library, 1D arrays - Generator L1S, 2D arrays - Generator A2S, Segments and Exporting Parameters sections of the online documentation.

8 Create your first 1D array

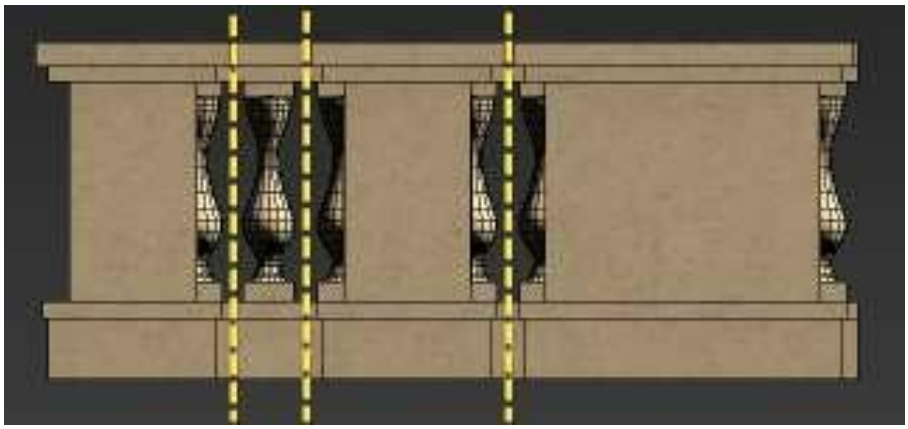
Having learned how to use and adapt the library, it's time to start creating your ownstyles. To do this it's important to have a good understanding of how base objects, segments and generators interact. In this section we'll learn how to use an L1S generator to create a low-poly background building.

To get the assets required to follow this chapter, download and open *l1s-exercise.max*.

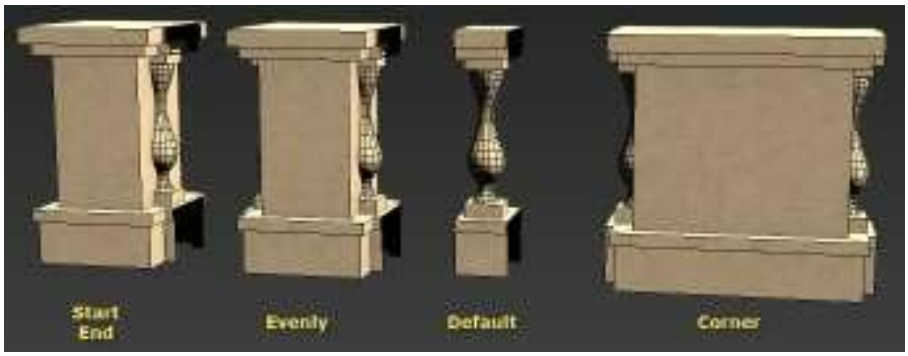
8.1 Creating geometry for an L1S Generator

The first thing to consider when creating a new RC style is how the individual pieces of geometry fit together. In Chapter 4 we examined the five parts of a one dimensional array, and when creating the meshes for segments it's important to think in terms of how they will fit into this system. You may find it is helpful to model a small part of the style as a single object and then break it apart using 3ds Max's Slice modifier, doing this will ensure that the resultant segments fit together correctly. Illustrated below are the pieces that you will use for the exercises in this chapter. The Walls and Balustrades were created using the Slice-modifier technique.

Balustrades



Balustrades model before slicing - yellow lines indicate slice planes

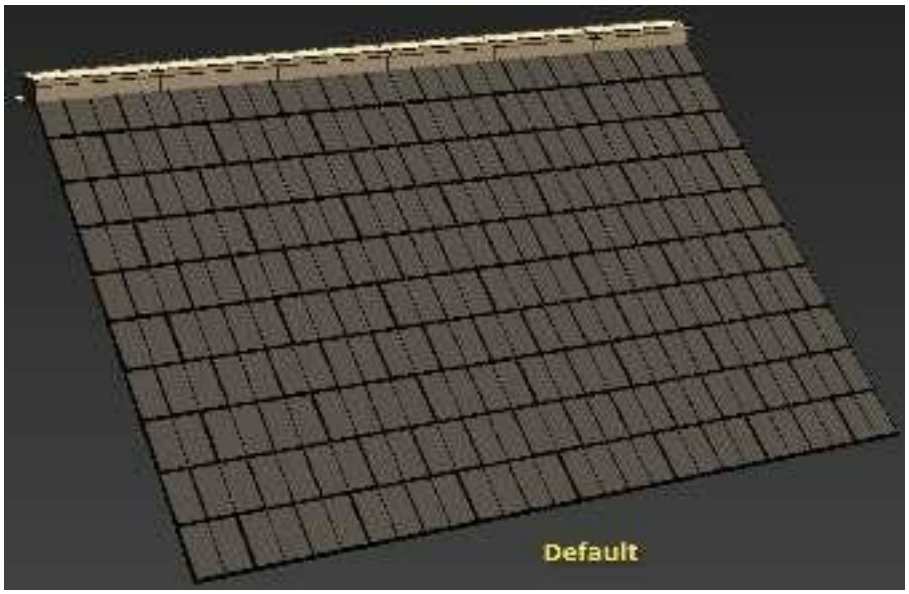


Balustrades model after slicing.

Walls

Walls after slicing.

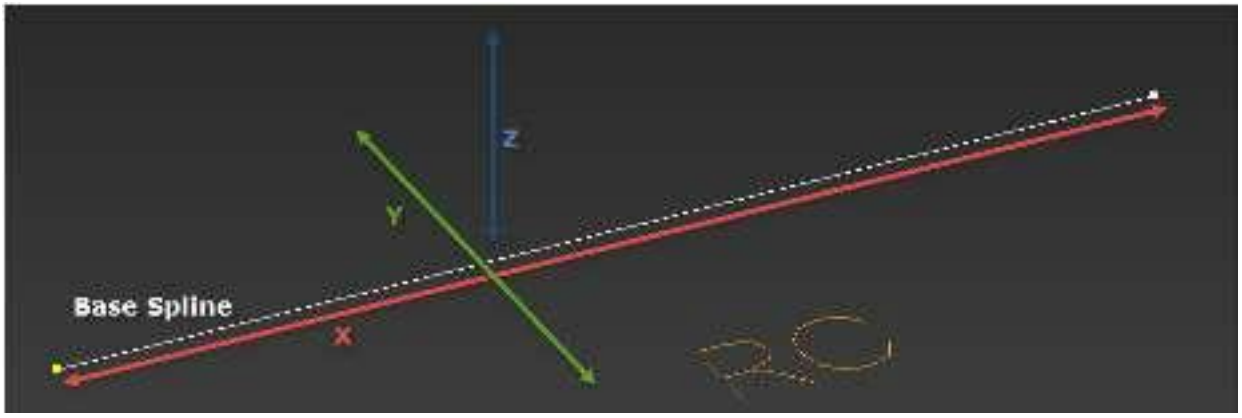
Roof



The roof using only one segment

Pivots

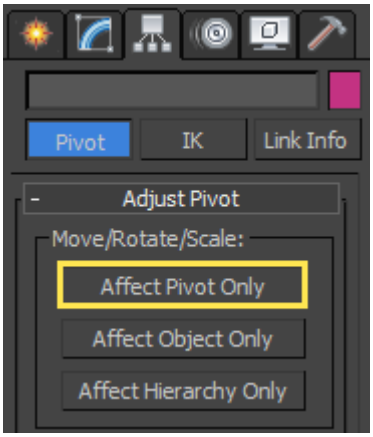
The geometry's pivots will be aligned with the coordinate system used by the RailClone object. In an L1S array, and when creating the RailClone object in a Perspective or Top viewport, the axes are oriented so that the X Axis follows the length of the spline, the Y Axis runs perpendicular to the spline, and the Z Axis is vertical, matching the world's Z axis and also the RailClone object's local orientation.



In order for geometry to be correctly aligned in the array, it may occasionally be necessary to adjust the orientation of the local axis. To do this:

1. Select the Geometry object.
2. Go to 3ds Max's **Hierarchy Panel**.

3. Click **Affect Pivot Only.**



4. Rotate the Gizmo to the correct orientation.

As well as orientation, another consideration is the position of pivot points. In order for segments to snap together correctly, it's possible to align the segments on the base object using the local pivot point of the source geometry. This is particularly true on the Y and Z axis, and less so on the X where RailClone's automatic mode is often more useful. In the examples on this page we will use the pivot point to position the all the segments on the Y axis and ensure they snap together correctly, to make this possible all of the pivot points are aligned on the X and Y axes as illustrated in the image below.

Material IDs and UVW Coordinates

By default RailClone retains the the UVW coordinates and Material IDs of the source geometry. It does not automatically create multi-sub object materials or reassign IDs. This means that when creating geometry it is important to ensure that the Material IDs would work as if the individual components were a single object with a single material (which can be a mult-sub), since this is how they will appear to RailClone.

8.2 Exercise: Creating A Rule-Set

As you saw in Chapter 3, the L1S generator has 5 inputs: Start, End, Corner Evenly, and Default.

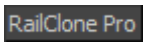
In this section we'll create an example that uses of all of these. It's important to note that it isn't necessary to use all 5 inputs, many styles may only require a selection of the available inputs. When an input is omitted the array is completed using Default segments.

Creating a RailClone Object

Once the geometry has been prepared, we're ready to create a new style. To do this we need to add a new empty RailClone object and open the Style editor:

1. Go to 3ds Max's **Command** > **Create** > **Geometry** panel and change the category to **iToo Software**.

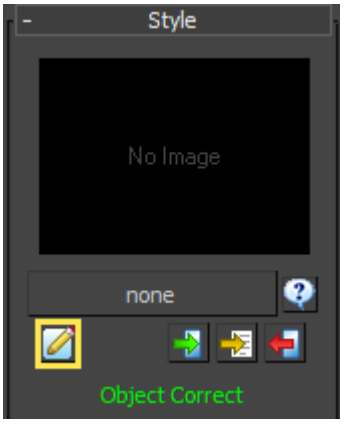
2. Click



.

3. Click and drag in the **Top** or **Perspective** viewport to create a RailClone icon.
4. Open the **Style** rollout and click

to open the **Style Editor.**



5. The Style Editor will open.

Using the Style Editor

Before creating your first style, take a few moments to learn how to interact with the style editor. As well as the essentials, you will learn a few lesser known commands that can make working with RailClone much faster.

To add a new node to the Construction view

Adding a new node

- Identify the node type in the **Items** list on the right hand side of the style editor, select and drag it into the **Construction View**.

To rename a node

- Select the node in the **Construction View**
- In the **Properties Editor** type a new value in the name field.

To connect nodes

Connecting nodes

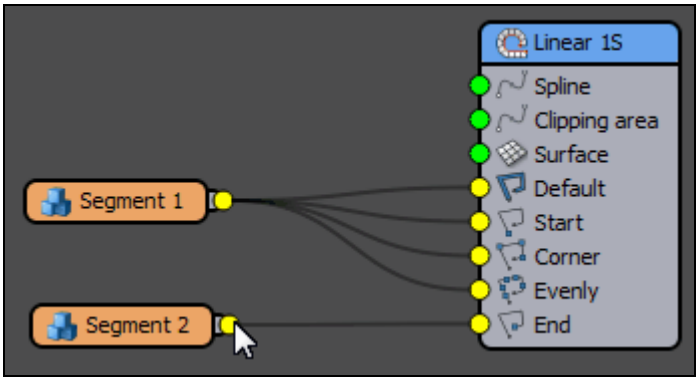
- Click on the output of a node, drag the mouse (you'll see a wire following the mouse pointer) and release over the input of another node.

To connect multiple nodes

Multiple nodes can be attached to Compose, Sequence, Random, and Selector operators. To do this:

1. Select the nodes you wish to attach.
2. Drag any of the node's outputs to the operator's input.

To swap the connections for two nodes



- Drag the output of a node 1 to the output of node 2. Their connections will swap.

To disconnect nodes

- Click on the output slot of a node and drag to a empty area of the **Construction View**. Any nodes connected to the slot will be released.

To collapse and expand nodes

Collapsing and expanding nodes

- To collapse a node, click on the small upward facing arrow on the right of the title bar
- Once a node is collapsed, to expand it again click on the small downward facing arrow on the right of the title bar

To automatically select a node's children

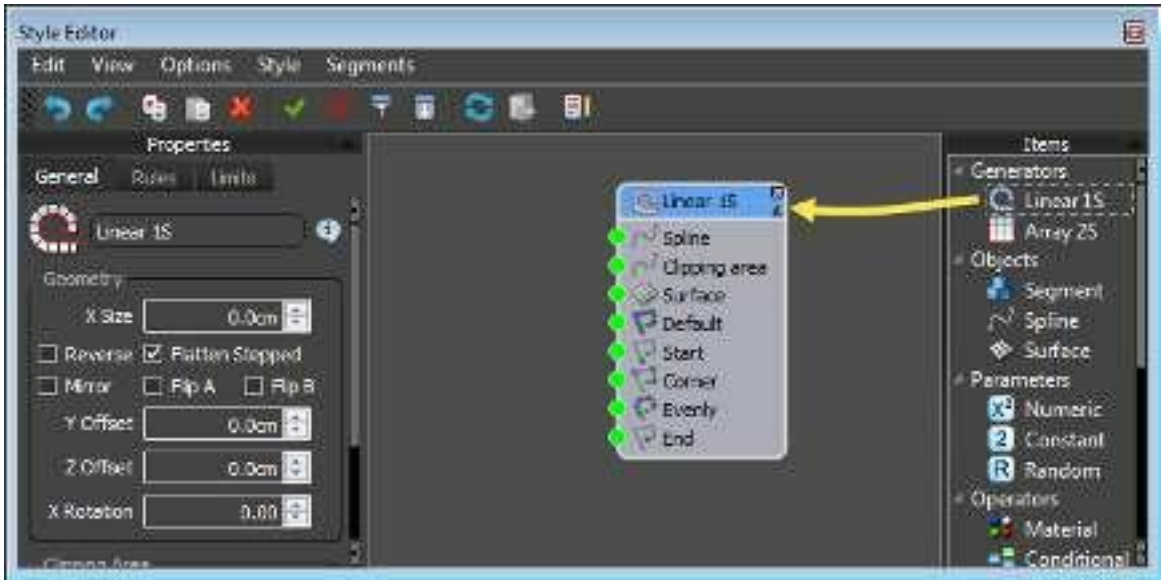
- Hold down **Alt** and **Click** on any node to automatically select its children.

Creating a Generator

Think of a generator as the brain of the style, it creates the rules that tell RailClone how to construct the array. A style needs at least one generator but is not limited to this, you can combine as many as necessary even mixing L1S and A2S generators to rapidly achieve sophisticated results. By the end of this example you'll create a total of 3 L1S generators; one for the façade, one for the balustrade, and one for the roof.

To create a generator for the façade, follow these 2 steps:

1. With the Style Editor open, drag a **L1S Generator** from the **Items List** to the **Construction View**.



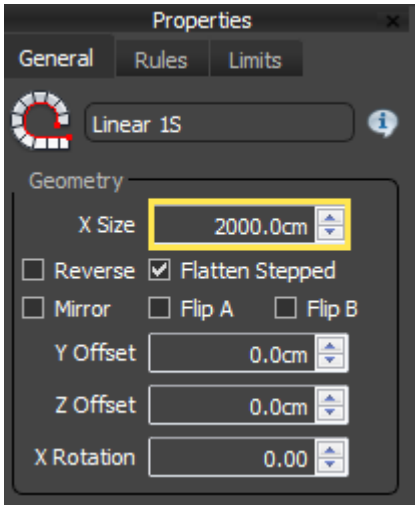
2. As we're going to add 2 further generators, let's name this one for easy identification. Select the node, go to the **Properties** panel and enter ***Façade L1S*** in the **Name** field.

Adding a Base Object

Now that you have an array, you need to specify how large it should be. There are two ways to set the dimensions of an array: you can either specify a distance value in scene units, or use a spline. To set a value:

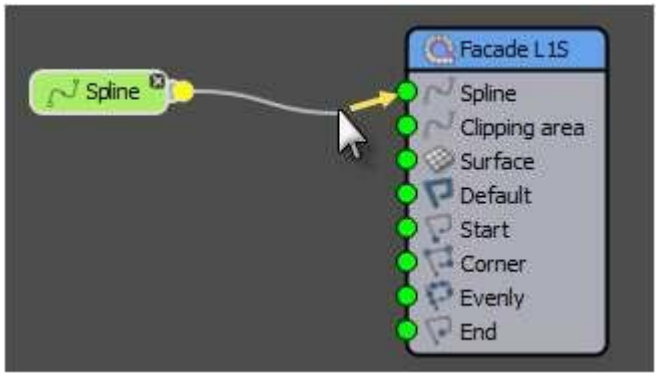
1. Click on the **Generator** node.
2. Go to the **Properties** panel.

3. Enter a distance value for the **Properties** > **Geometry** > **X Size** parameter.



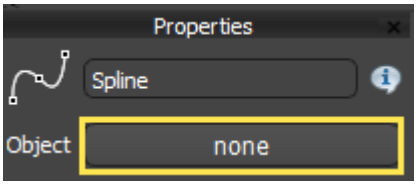
Alternatively, as in this example, you can use a spline to determine the size and path of the array. In RailClone terminology we call a spline used in this way a **Base Object**. When using dimensions the array is always straight, but when using a spline, the array can follow and deform along complex curves. To add a spline Base Object to the generator created above, follow these steps:

1. Drag a **Spline Object** from the **Items List** to the **Construction View**.
2. Click and hold on the **Output** of the Spline node and drag a wire to the Spline **Input** of the generator.



3. We now need to tell RailClone which spline in the scene to use. You can select the spline in one of two ways:
   From inside the style Editor
   a. Select the **Spline** Node and go the the **Properties Panel**.

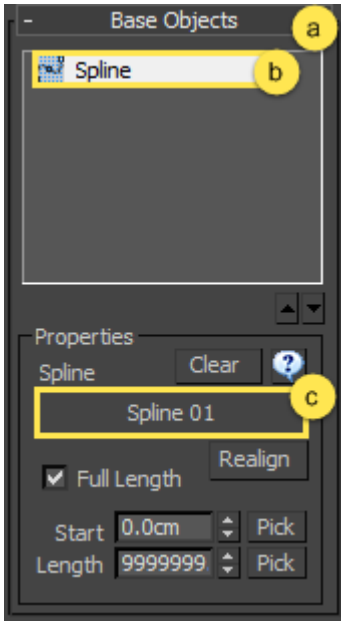
b. Click the **Object Picker** button



c. Click the spline in the scene called ***Spline 01***.

From the Modify Panel

When an object node is created in the Construction view, a corresponding item is added to the Base Objects rollout in the Modify Panel. This allows you to apply the style to a spline without having to open the style editor. To do this:



1. a. Open the **Base Objects** rollout in the **Modify Panel**.
   b. Select ***Spline*** from the Base Objects list
   c. Click on the Spline picker button and select the spline called ***Spline 01*** from the scene.

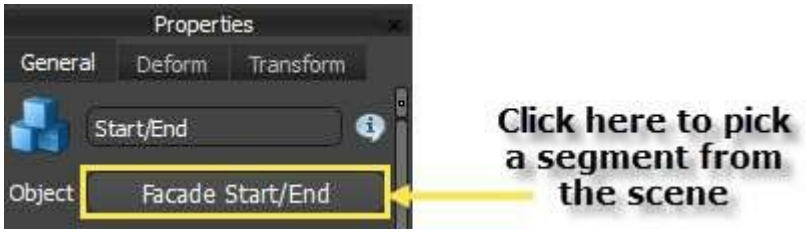
Adding Segments

With a generator path successfully created, the final ingredient is geometry. In RailClone terminology, an individual mesh object added to a style is called a **Segment** and they are created in much the same way as base objects. To create a segment you first add a Segment node to the construction view and and then assign an object from the scene. To add the first segment, follow this procedure:

1. Create a new **Segment Object** (

)by dragging from the **Items List** to the **Construction View**.

2. Go to the **Properties Editor** and click on the **Object** picker.

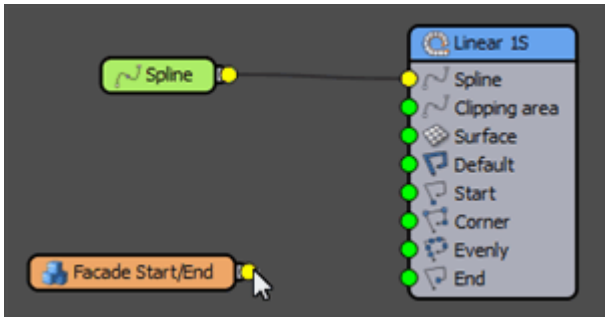


3. Select the geometry you wish to use from the scene, in this case we'll pick ***Facade Start /End/Evenly***.

The Start and End input

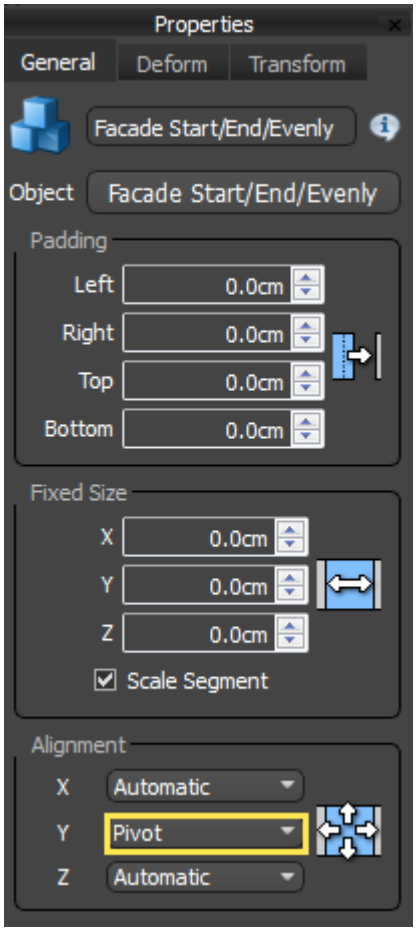
Now that we have a segment, we need to assign it to a part of the array. In this section we'll add the segment to the Start and End inputs. Wiring a segment to these inputs creates geometry at the start and end of the path, as defined by the spline's original vertex numbers.

1. Click and hold on the Segment node's **output** socket, then drag a wire to the Generator's **Start** input socket.

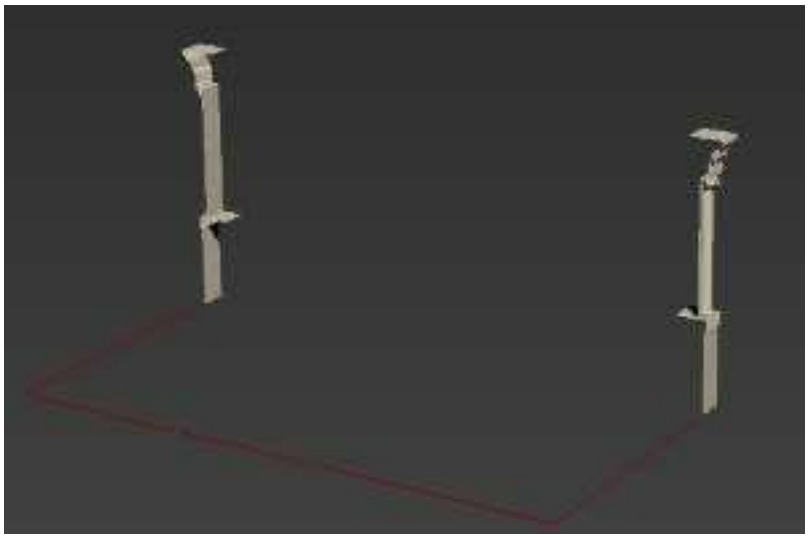


2. Repeat the action to attach the Geometry node to the Generator's **End** input.

3. Click on the Geometry Node, go to the Properties panel and change the **Alignment > Y** to **Pivot.** As mentioned in the geometry section above, to ensure that the segments fit together correctly all the segments used in this style will use **Alignment > Y = Pivot**.



4. You will now have a segment at the start and end of the Spline:



Adding multiple segments to an input

Sometimes you need to assemble several segments for each position, not only one piece. In this case, you can use a "Compose" operator, to join several elements in an unique segment. We examine this in more detail in Chapter 11.

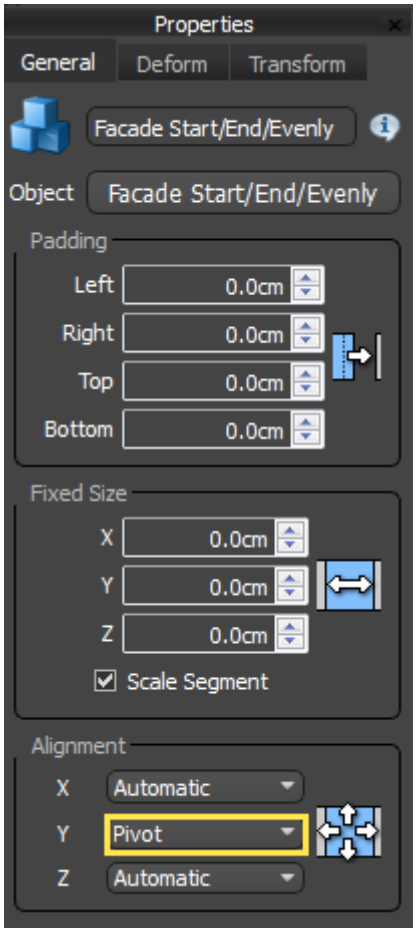
The Corner input

To create corners, we wire a segment to the "Corner" connector on the generator's nod e. Corner geometry appears only on vertices of a specified type found on the path. By default only Corner or Bezier-corner vertices are used, but this can be changed from the Generator's properties. In this example we'll retain the default settings.

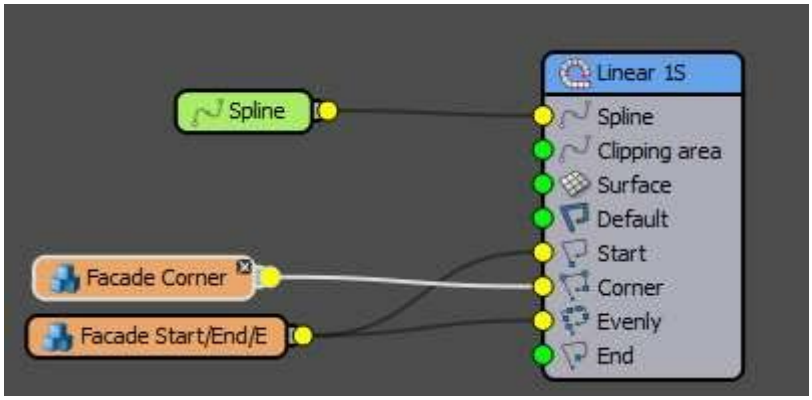
1. Create a new **Segment Object** (
   

)by dragging from the **Items List** to the **Construction View**.
2. Go to the **Properties Editor** and click on the **Object** picker.
3. Pick ***Facade Corner*** from the scene.

4. Still in the new Segment node's **Properties**, change **Alignment** > **Y** to **Pivot.**

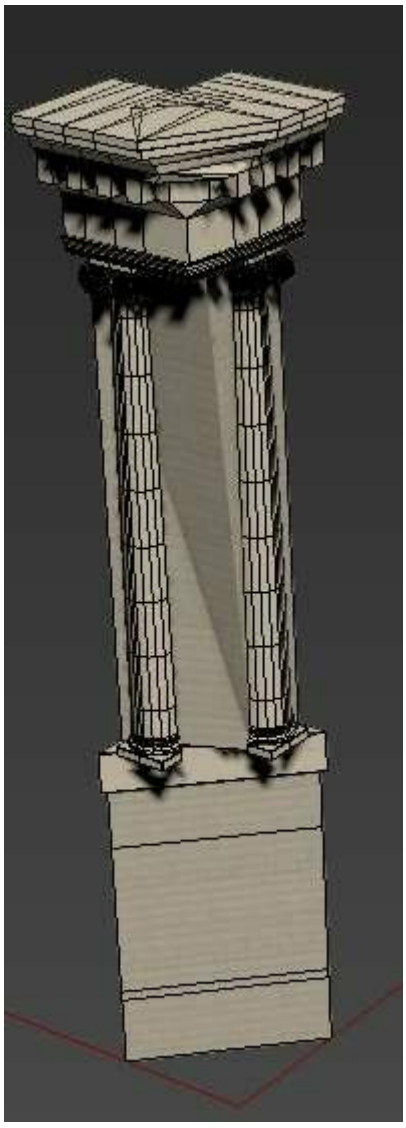


5. Wire the Segment to the Generator's **Corner** input.

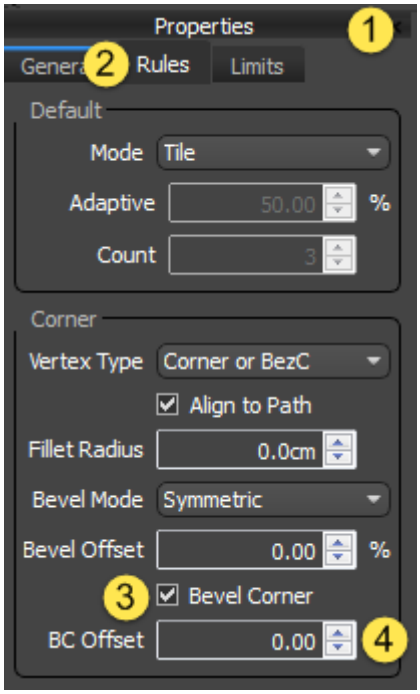


Bevelling corners

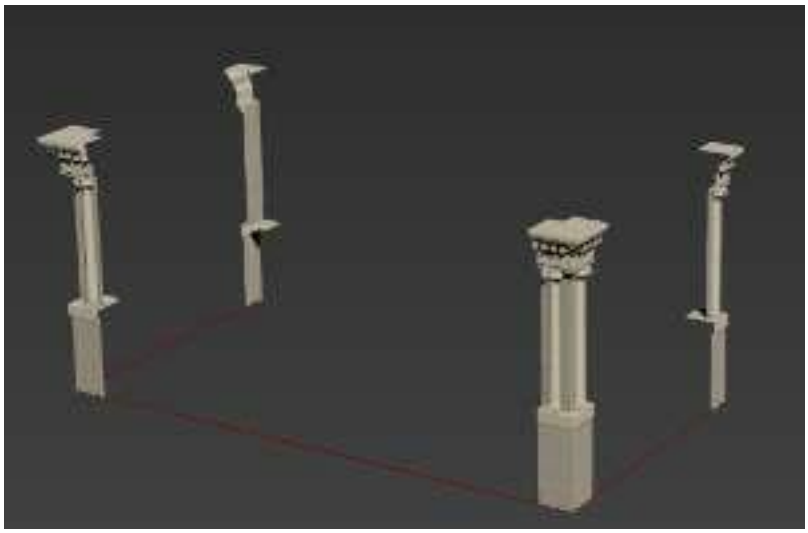
By following the previous steps, the corners appear, but they look a little strange:



This is because by default corners will try to deform to follow the path, so the geometry is attempting to bend around a 90 degree corner. Instead we want the corner to be **bevelled**. To do this make these changes:



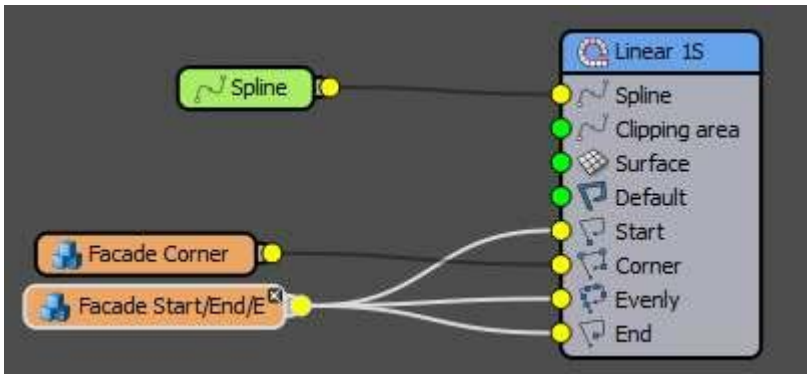
1. Select the **Generator** and go to the **Properties** panel.
2. Click the **Rules** tab.
3. Activate the **Bevel Corner** checkbox.
4. The segments are now sliced on the corner. You can adjust this further using the **BC Offset** value which allows you to add an offset to the Corner segments. Positive values pull the segment towards the corner, and negative values move the segments away from the corner. For this example enter a value of **35%** to ensure that the Corner segments bevels in the correct place.
5. You have now successfully added Start, End and Corner segments to the array:



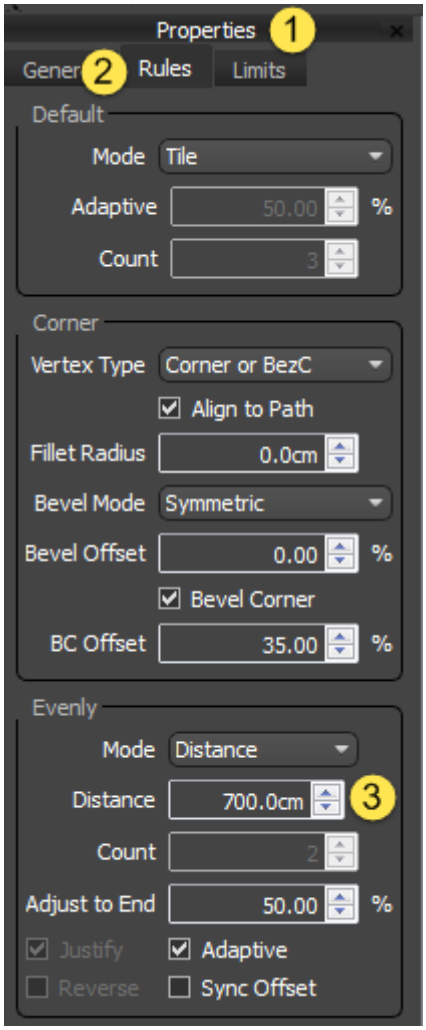
The Evenly input

This construction rule lets you to add regularly spaced segments along the path between Start, End and Corner segments. In this example you'll reuse the existing segment named *Facade Start /End/Evenly*.

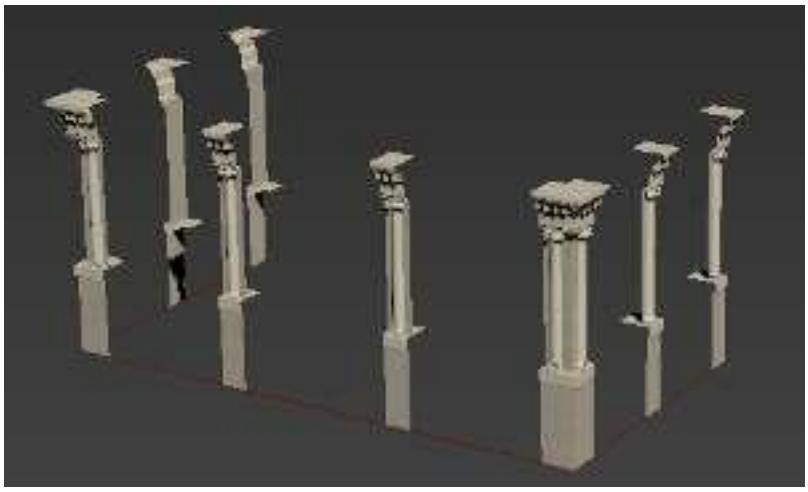
1. Wire the segment named ***Facade Start/End/Evenly*** to the Generator's **Evenly** input.



2. To change the spacing between Evenly segments, go to the Generator's **Properties** (1). Click on the **Rules** tab (2) and edit the **Evenly** > **Distance** value (3). For this example enter a value of **700cm**.



3. You have now successfully added Start, End, Corner and Evenly segments to the array. There's only one more input left to add - Default.



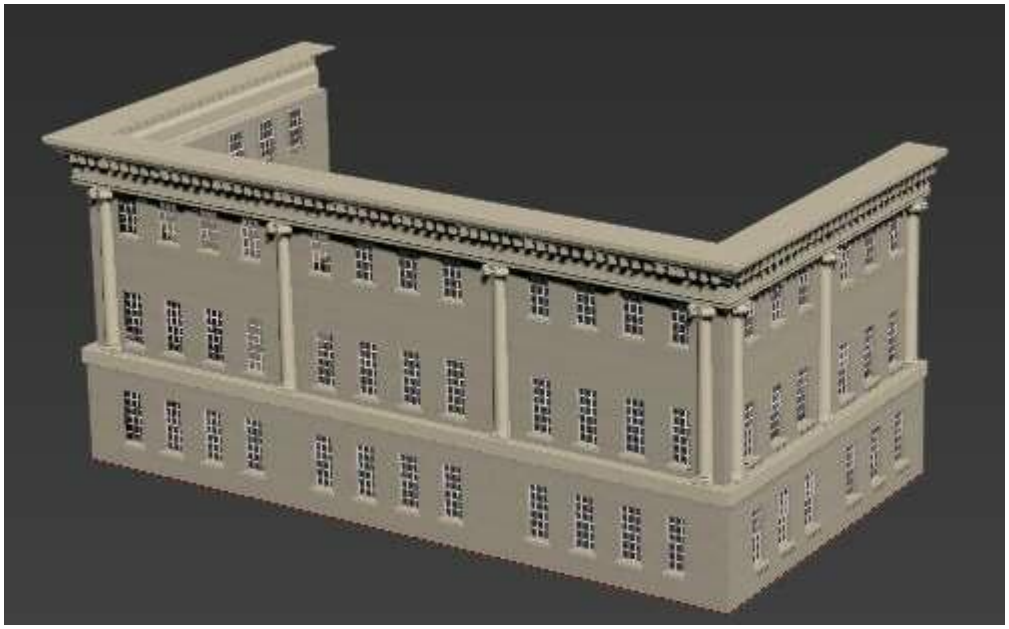
The Default input

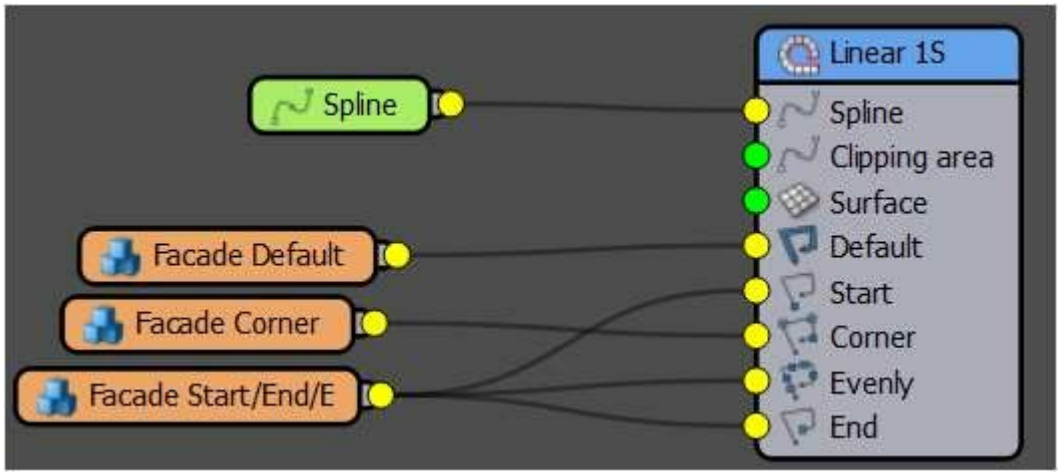
The segment used in the default input fills in between the Start, End, Corner, and Evenly segments. To add a Default segment:

1. Create a new **Segment Object** (
   

)by dragging from the **Items List** to the Construction View.
2. Go to the **Properties Editor** and click on the **Object** picker.
3. Pick ***Façade Default*** from the scene.
4. Still in the new Segment node's **Properties**, change **Alignment** > **Y** to **Pivot.**
5. Wire the new segment to the Generator's **Default** input.

6. You have now successfully added start, end and corner and default segments to the array and completed the façade. In the next sections you add additional generators to create a balustrade and roof.

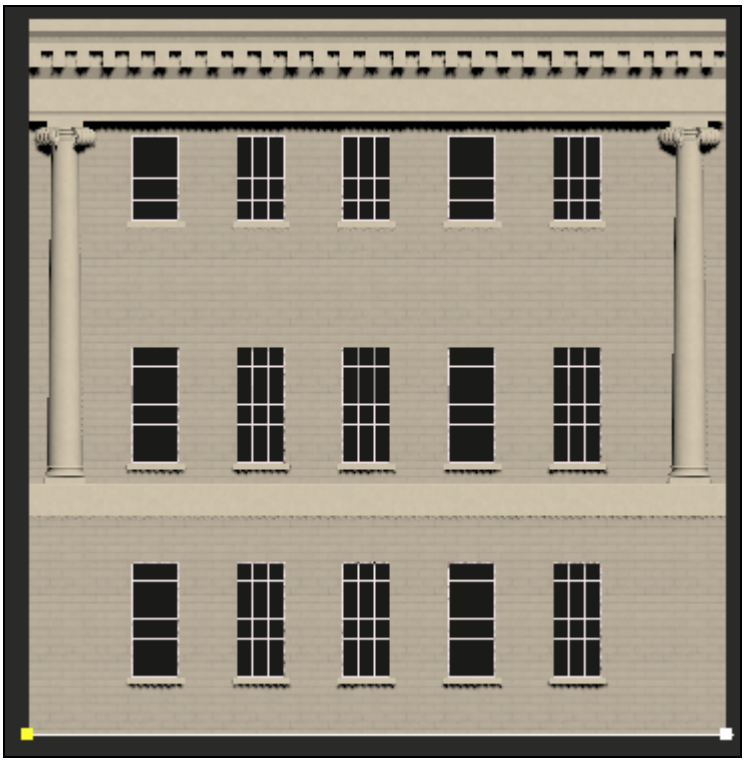




Default Modes

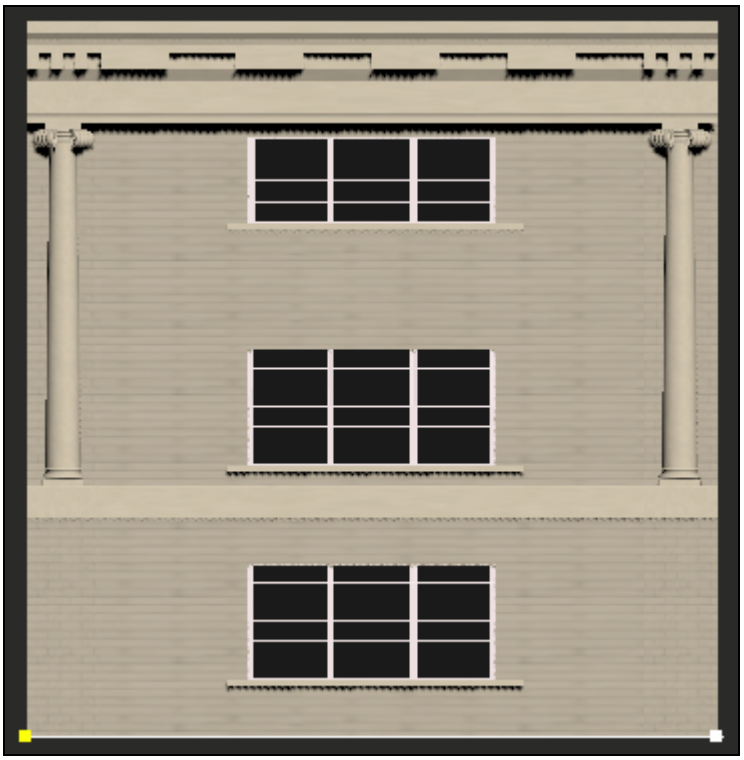
Though in this example the default segments simply tile along the path, RailClone has 3 additional modes for calculating how to fill a section with Default geometry:

Tile (Default)



The default segment are replicated along the path, slicing the geometry if necessary.

Scale

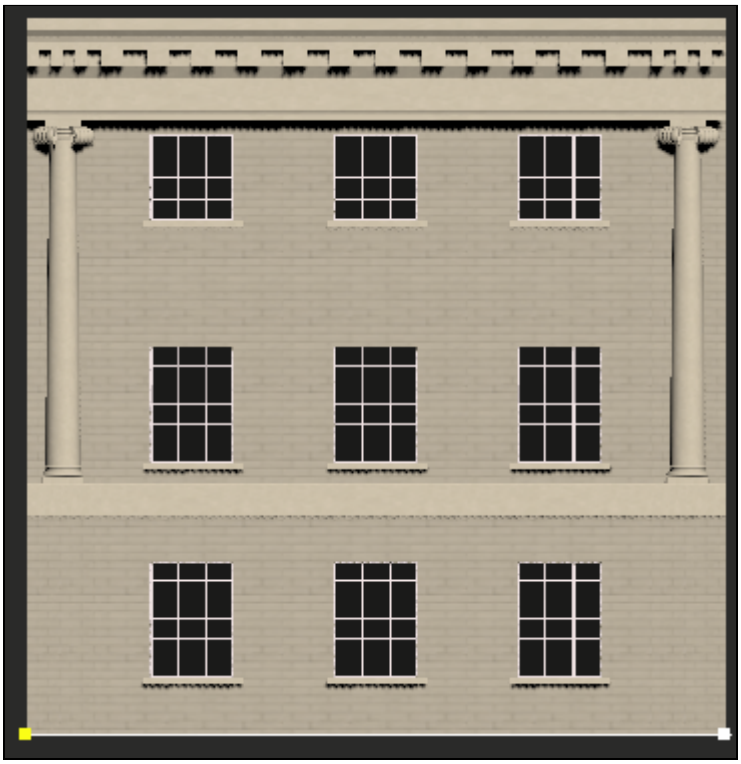


The default segments are scaled along the X axis to fit the distance between the initial and final positions of the path section. A section is defined by the Start and End points of the spline unless an X-Evenly or corner segment is used to subdivide the path.

Adaptive.

Adaptive mode calculates the number of whole segments that would fit along a spline section then scales them all on the X axis to fill the available space. This mode generates complete segments, with no clipping at the start or end of a spline segment. When Adaptive mode is active the bevel settings are unavailable.

Count Mode.



Scales a fixed number of segments to fit between the start and end positions of a path section. A section is defined by the start and end points of the spline unless a X-Evenly or corner segment is used to divide the path. When Count mode is active the bevel settings are unavailable.

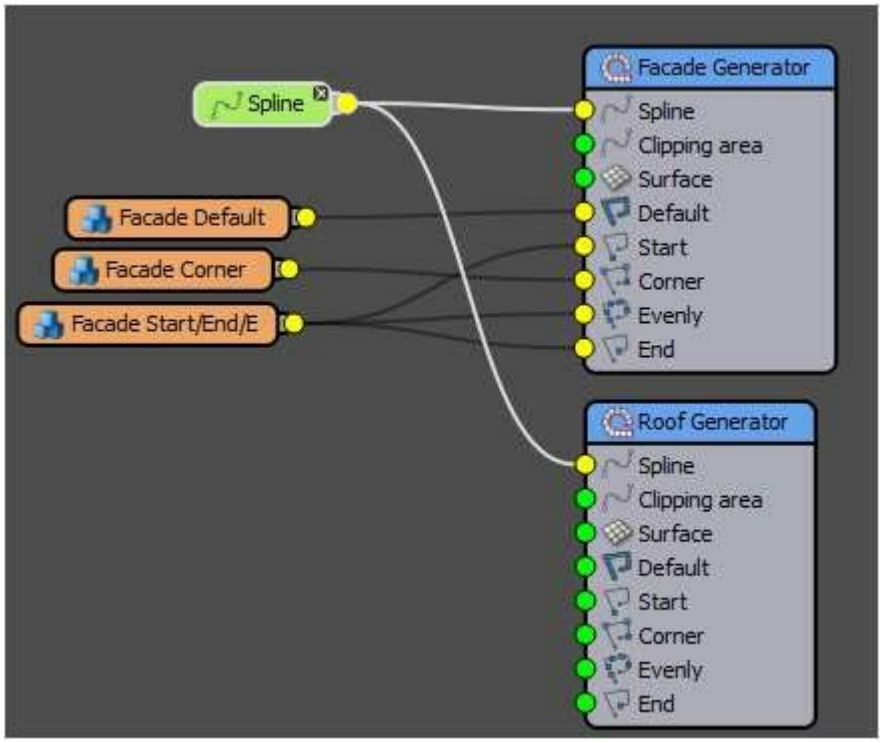
8.3 Exercise: Using multiple generators

As styles become more sophisticated it is often much easier to use multiple generators. In this example we'll add a 2 more generators to the existing style, one for the roof and a third for the Balustrade.

Adding the Roof.

1. Make sure you continue with the existing Building style, and open the **Style Editor**.
2. Add a second **L1S Generator** (
   

) by dragging it from the **Items List** to the **Construction View**.
3. Rename the generator ***Roof Generator***.
4. Wire the existing **Spline** node to the roof Generator's **Spline** input. By reusing the spline in this way, you can be sure that both generators follow the same path.



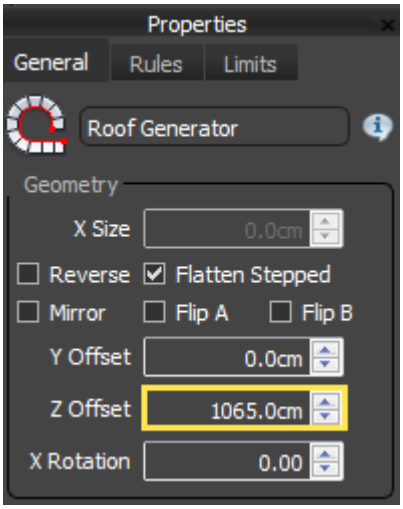
5. Create a new **Segment Object** (
   

)by dragging from the **Items List** to the **Construction View**.
6. Go to the **Properties Editor** and click on the **Object** picker.
7. Pick the object named ***Roof*** from the scene.
8. Still in the new Segment node's Properties, change **Alignment > Y** to **Pivot.**
9. Wire the new segment to the roof Generator's **Default** input.

You will now have a roof in the scene, but it's at the base of the building instead of above the existing façade.

To resolve this you can use the Generator's **Z Offset** parameter to move the roof into the correct position. To do this:

1. Select the roof Generator.
2. Go to the **Properties** panel and enter a value of **1065cm** for the **Z Offset** parameter.



3. The roof is now in the correct position. In the next section we'll add the balustrade, using the same Z offset value to ensure its position matches the roof.

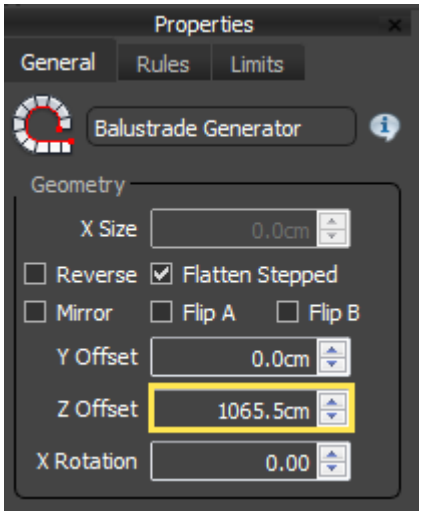
Adding the Balustrade

1. Add a third **L1S Generator** (

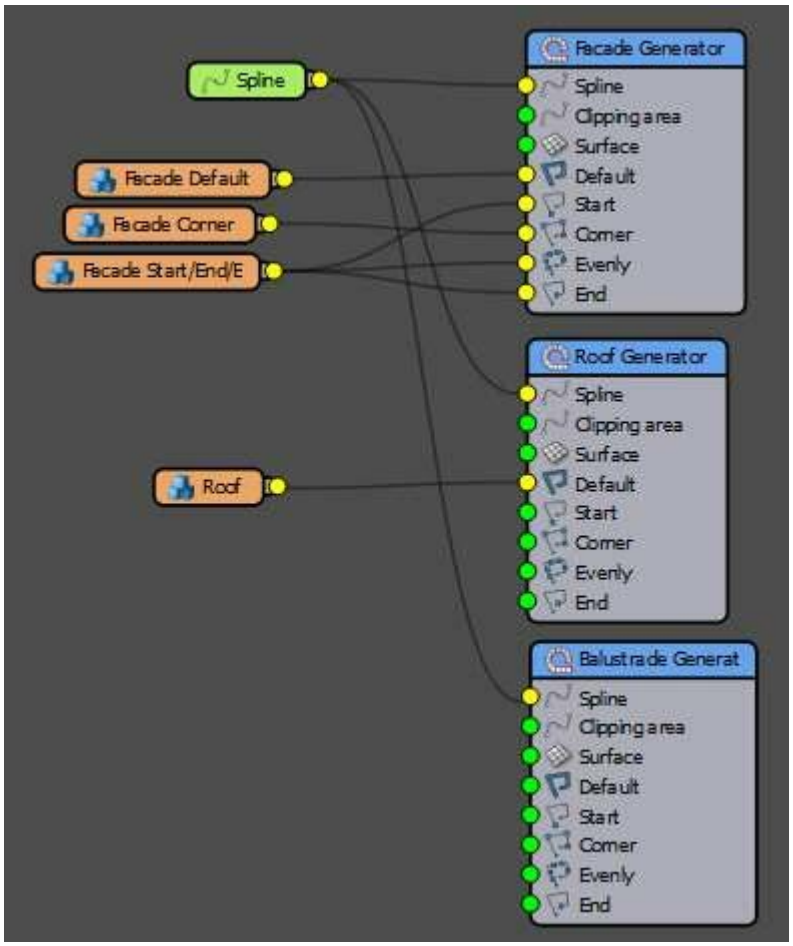
) by dragging it from the Items List to the Construction view.

2. Rename the generator ***Balustrade Generator***.

3. Go to the Properties panel and enter a value of **1065cm** for the **Z Offset** parameter.

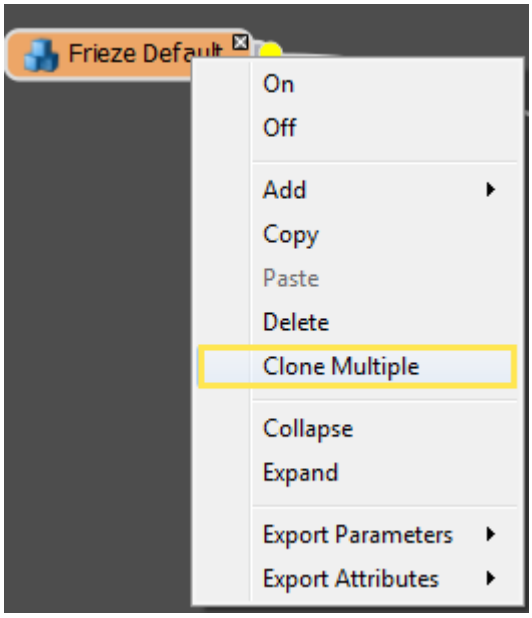


4. Wire the existing Spline node to the balustrade Generator's **Spline** input. All 3 generators are now sharing the same spline.

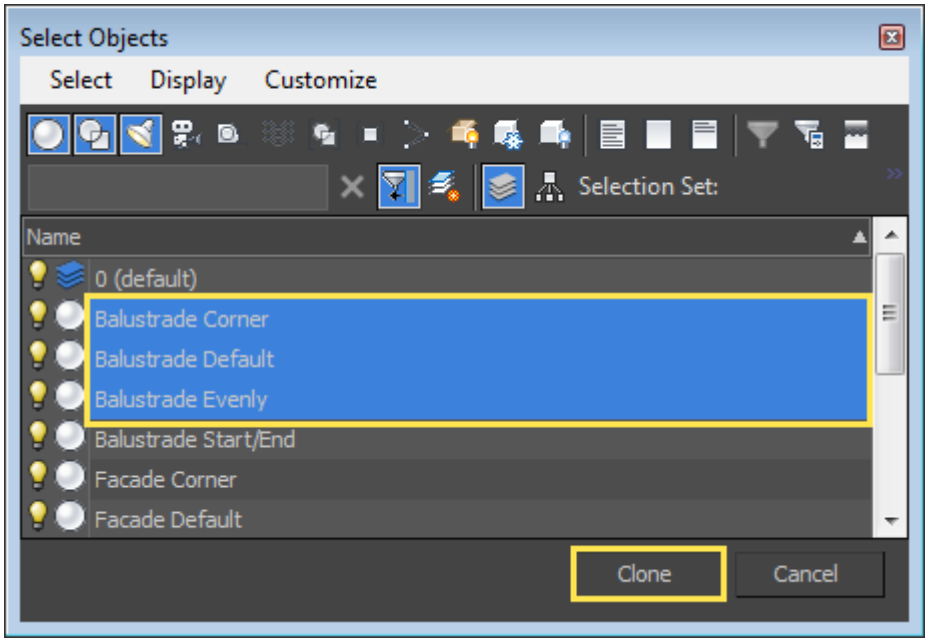


5. Create a new **Segment Object**.
6. Go to the **Properties Editor** and click on the **Object** picker.

7. Pick the object named ***Balustrade Start/End*** from the scene.
8. Still in the new Segment node's Properties, change **Alignment > Y** to **Pivot.**
9. To save time we can use this node as a template to quickly add the other geometry required. To do this. right click on the new Segment and select **Clone Multiple.**

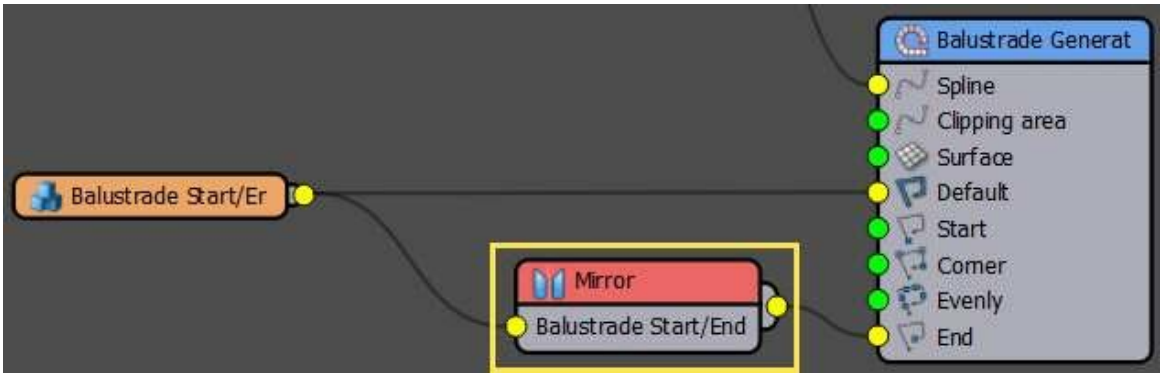


10. A **Select Objects** window opens. Pick ***Balustrade Default***, ***Baluststrade Evenly***, and ***Balustrade Corner*** then click **Clone.** RailClone will generate a new Segment node for each of the selected items.

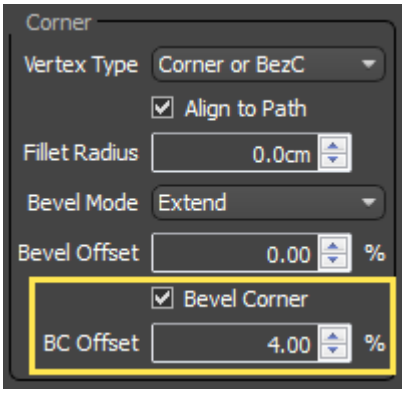


11. Wire ***Balustrade Start/End*** to the balustrade Generator's **Start** input.

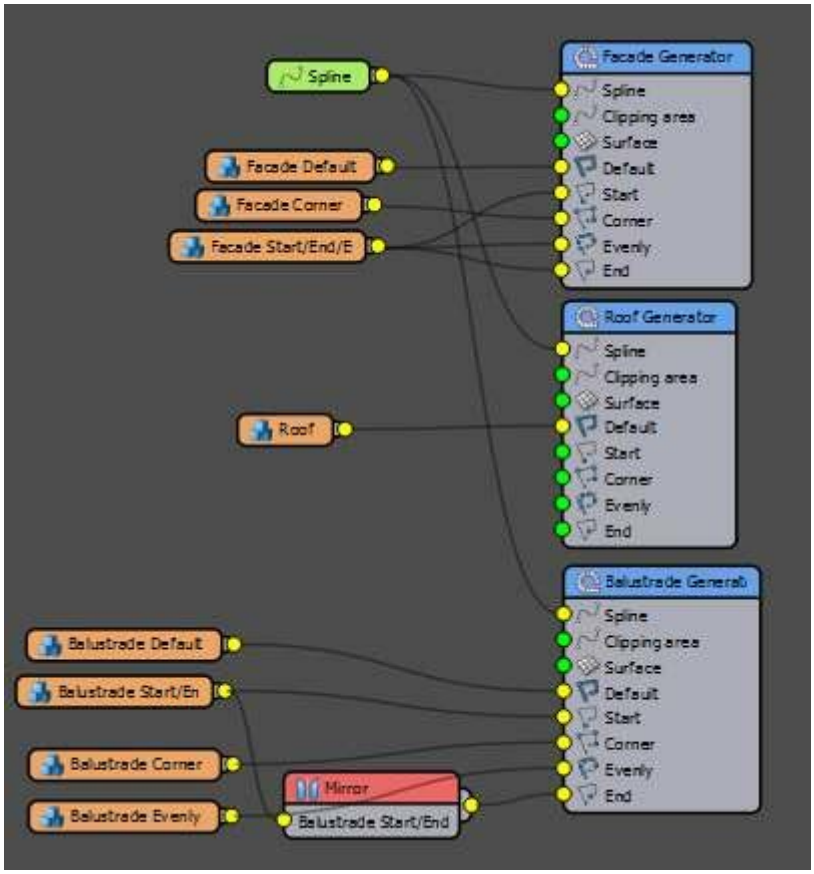
12. We want to reuse this segment in the End input, but it needs to be mirrored first. Instead of creating new geometry and adding an extra segment you can use a Mirror operator to flip the segment directly inside of RailClone. To do this, drag a new **Mirror** operator () from the **Items List** to the **Construction View**. Wire ***Balustrade Start/End*** to the Mirror node's **input**. Wire the Mirror node's **output** to the Generator's **End** input. (we will discuss operators in more detail in Chapters 10 and 11)



13. Wire ***Balustrade Default*** to the balustrade Generator's **Default** input.
14. Wire ***Balustrade Evenly*** to the balustrade Generator's **Evenly** input.
15. Wire ***Balustrade Corner*** to the balustrade Generator's **Corner** input. Like the façade example above, the cor ner settings now need to be adjusted.
16. Click on the balustrade **Generator** and from the **Properties** panel, activate **Rules > Corner > Bevel Corner** and set the **BC Offset** value to **4%**.



This completes the building style. The complete node tree should look like this:



Once a style is created it's simple to reuse and re-purpose. Try cloning the object and swapping the base object by picking ***Spline 01*** and ***Spline 02***. RailClone deforms the geometry to follow the curves and corners of any path.

8.4 Related Tutorials

Masonry Wall

The Masonry Wall tutorial is good accompaniment to this chapter. It is intended for new users of RailClone, and provides a thorough introduction to some of the most commonly used features. By completing this exercise you will be able to use the built in library, understand the style editor, and easily create new styles.

Power Lines

This tutorial explains in more detail the Default modes and walks you through the steps necessary to create power lines.

Stadium Tutorial Part 2,3, and 4

The stadium tutorial uses a number of L1S arrays to create advertising hoarding, seating and the crowd. These expand on the fundamentals discussed here to introduce randomisation, controlling material IDs, exporting parameters and using many useful operators.

Create a seaside promenade

In this tutorial we explain how to use RailClone to create your own 1D and 2D arrays to recreate a Seaside Promenade image. This tutorial builds on the foundations in this chapter and explores in detail how to use styles from the library, create new styles, use operators, edit padding and introduce 2d arrays.

Documentation

For more information please see the 1D arrays - Generator L1S, Path Base Objects, Style Editorand Segments sections of the online documentation.

9 Create your first 2d array

The 2D Array , or A2S generator, features 12 possible inputs, making it an incredibly powerful and versatile tool for creating parametrised objects including façades, floors, ceilings, pavement, tiles, structural steel, cladding and much much more. Before creating your first style, take a few minutes to get reacquainted with the different parts of the A2S Array. The exercise on this page makes use of all 12 inputs, but often you'll only need a selection of these to get the correct result.

This generator can be confusing for new users, but once you've understood the L1S generator, the A2S generator becomes a lot easier. The best way to understand the nuances of the A2S generator is to think of it as a stack of L1S generators. Internally, each row is treated as an independent 1D array, each with it's own start, end, corner and evenly segments. The A2S array uses 4 types of row: one at the **Bottom** of the array, one at the **Top**, several spaced **Evenly** on the **Y** axis, and a **Default** row that fills in whatever is left between the other 3. Each of these rows independently behaves exactly like an L1S generator, but the inputs are renamed. The animated gif and table below demonstrate the relationship between a stack of L1S arrays and the input names as they appear in the A2S generator

L1S Row	Start	Corner	Evenly	Default	End
Top Row	Top Left Corner	Inner Corner	X Evenly	Top side	Top Right corner
Y Evenly Row	Left side*	Inner Corner	X Evenly	Y Evenly	Right side*
Default Row	Left side	Inner Corner	X Evenly	Default	Right side
Bottom Row	Left Bottom corner	Inner Corner	X Evenly	Bottom row	Right Bottom Corner

* (or Y Evenly if extend to side is active)

9.1 Creating Geometry for an A2S Generator

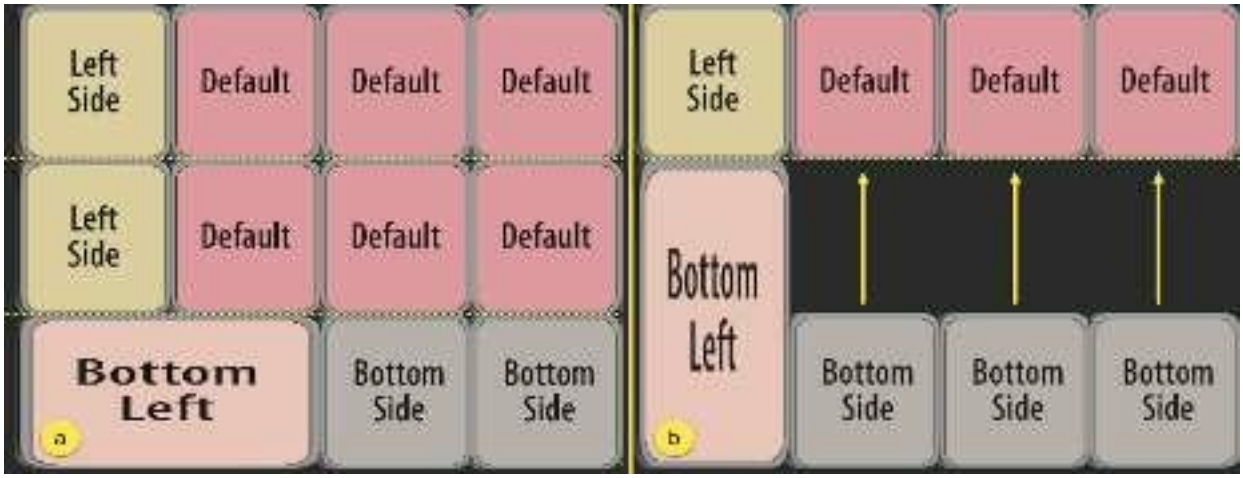
At the beginning of the previous chapter we discussed the importance of planning your geometry to ensure it fits correctly into the different parts of the array. The same rules apply, and are even more important when creating geometry for the A2S array. To recap, the best way to understand the A2S generator is to think of each row as a separate L1S generators, each once stacked on top of another, but all still essentially independent. This understanding is essential if you want to create geometry to populate an array that is intended to fit together perfectly without leaving any gaps.These are the 3 Golden rules for modelling for the A2S Generator.

Rule 1: Segments can be any X length

As we've seen, each Row is a separate L1S generator and is calculated independently, so there's no problem at all with using segments with unequal sizes on the X axis. Adjacent segments in the same row will simply move along the array to accommodate any size of geometry. Rows above and below remain unaffected and **no gaps appear**.

Rule 2: To avoid gaps, Segments should be the same Y length

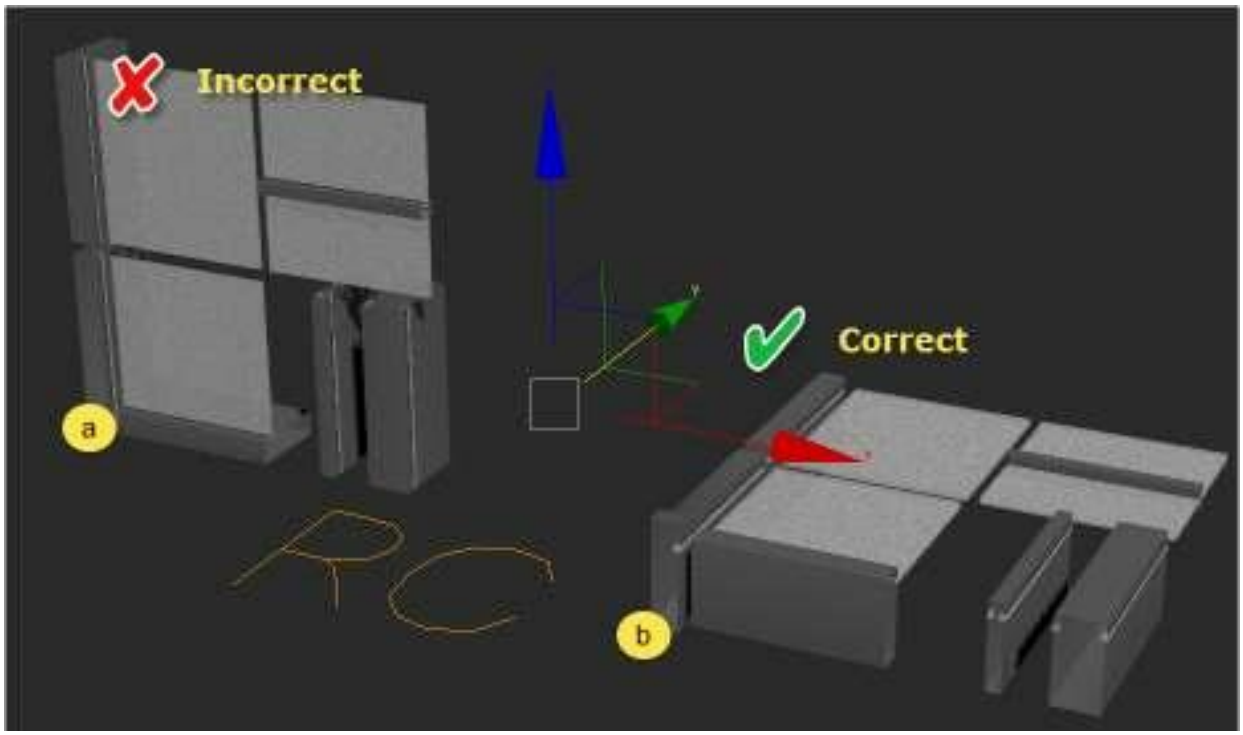
Because the A2S array is a stack of L1S generators, if you use Segments with unequal sizes on the Y axis in a row, the overall height of that row is determined by the tallest segment. The row directly above will be positioned **after** the tallest segment and **gaps will appear**.



a. The corner segment has been scaled to twice it's normal width, adjacent segments in the same row adjust automatically an no gaps appear | b. The corner segment has been scaled to twice it's normal height, the row above moves to accommodate the largest size and gaps appear in the array.

Rule 3: Align the source geometry's pivots to the X and Y axes

The 2d array is created along the X and Y axes of the RailClone object's local coordinate system, and segment's will be oriented to that their own axes align with this. It's very tempting, especially with vertical styles like façades and cladding, to align geometry incorrectly so that the vertical elements are on the Z axis. For example, the image below illustrates the segments used in the façade exercise on this page. The collection of segments marked **A** are **incorrectly** orientated so that there verticals align to the Z Axis. The group of segments marked **B** are **correctly** oriented. Their segments are aligned to the X and Y axis, matching the axes used by RailClone to construct the array.



9.2 Exercise: Creating glazing using a 2D Array

In this exercise you'll create a new glazing style from scratch that uses all 12 inputs of the A2S generator. By doing this you should gain a good understanding of the A2S generator.

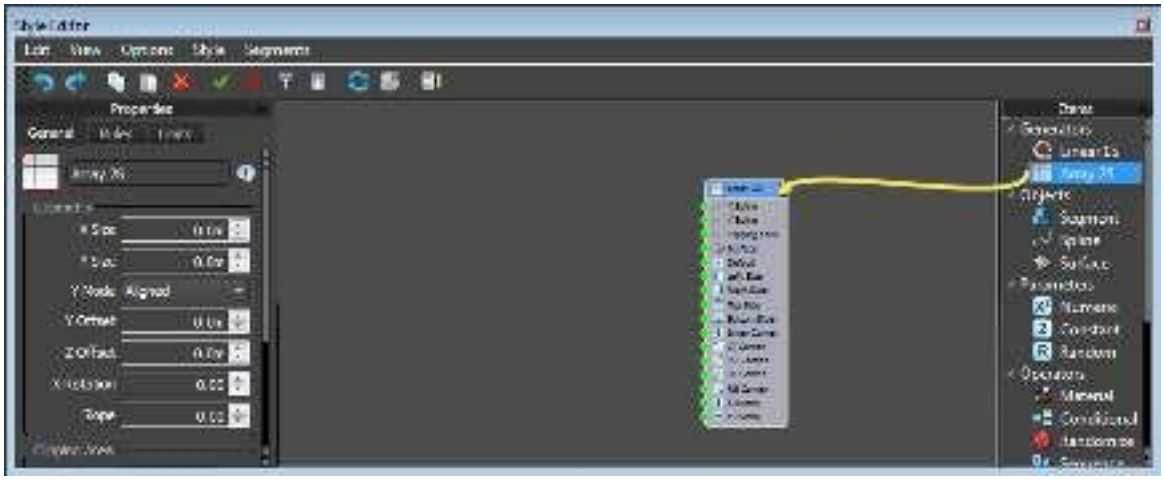
Creating a Generator

To create a generator for the Glazing , follow these steps:

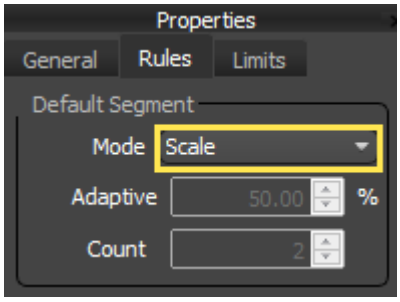
1. Open the file named *A2S_generator.max*. This scene contains all the geometry and paths required to recreate this style.
2. Create a new **RailClone object** and name if ***rc_glazing***.
3. Go to the **Modify** panel, open the **Style** rollout and click on 

   to open the **Style Editor.**

4. With the Style Editor open, drag a **Array 2S Generator** from the **Items List** to the **Construction View**.



5. Open the Generator's properties and set the **Rules** > **Default Segment** > **Mode** type to **Scale**.



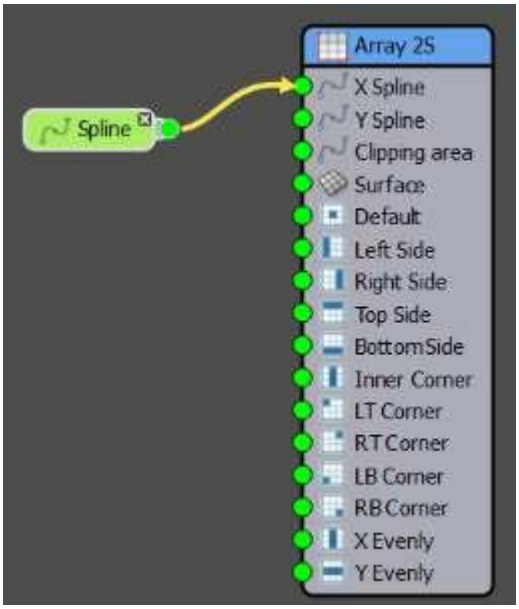
Adding Base Objects

Now that you have an array, you need to specify how large it should be. To create a an A2S array you need to set a distance for the **X** and **Y** axes. There are two ways do this, you can either specify a distance value in scene units, or use a spline. In this example you'll use a spline or the X axis and use a distance for the Y axis:

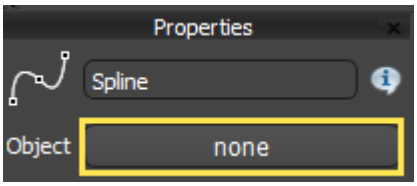
Add a spline for X Axis

1. Drag a **Spline Object** from the **Items List** to the **Construction View**.

2. Click and hold on the **Output** of the Spline node and drag a wire to the **X Spline Input** of the generator.



3. We now need to tell RailClone which spline in the scene to use. You can select the spline in one of two ways:
   Either from inside the style Editor...
   a. Select the **Spline** node and go the the **Properties Panel**.
   b. Click the **Object Picker** button



   c. Click the spline in the scene called ***Path Spline***.

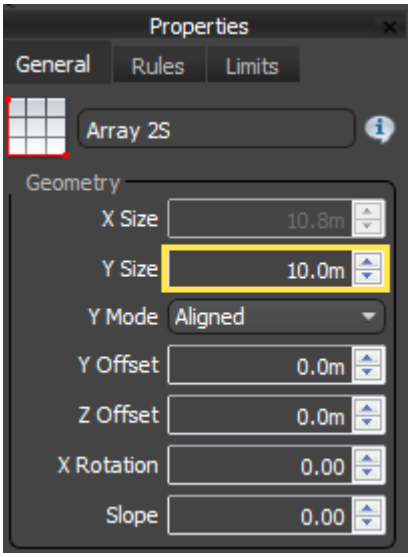
...or from the Modify Panel

1. a. Open the **Base Objects** rollout in the **Modify Panel**.
   b. Select ***Spline*** from the **Base Objects** list
   c. Click on the **Spline picker** button and select the spline called ***Spline 01*** from the scene.

Set a distance for the Y axis

1. In the **Style Editor**, click on the **Generator** node.
2. Go to the **Properties** panel.

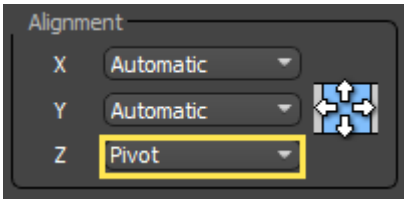
3. Enter a distance value of **10m** for the **Properties** > **Geometry** > **Y Size** parameter.



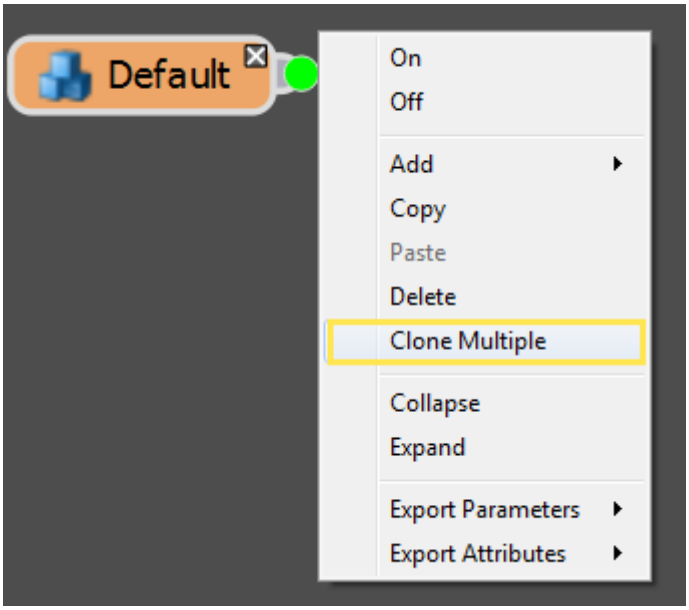
Importing Segments

The Glazing style is constructed from seven different segments. Adding these one at a time is time-consuming, so to speed things up we'll automate the process using RailClone's **Clone Multiple** command.

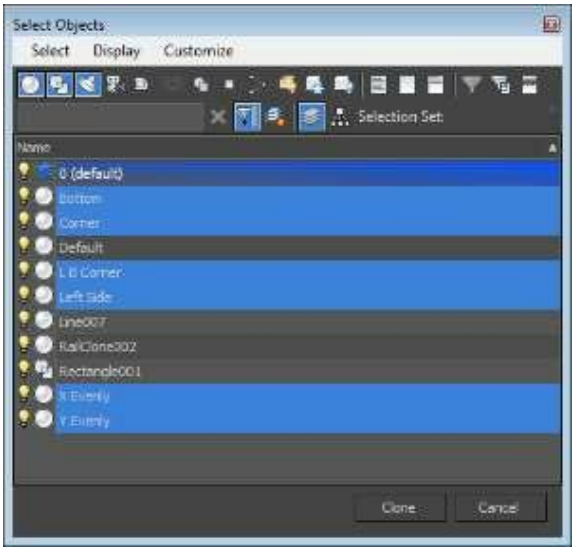
1. Create a new **Segment Object**.
2. Go to the **Properties Editor** and click on the **Object** picker.
3. Pick the object named ***Default*** from the scene.
4. Still in the new Segment node's Properties, change **Alignment** > **Z** to **Pivot.** This is to ensure that the glass and the seals and glazing bars align correctly.



5. We can use this node as a template to add the other geometry object. These will inherit the Alignment properties set in the previous step. To do this. right click on the new Segment and select **Clone Multiple.**



6. A **Select Objects** window opens. Pick ***Bottom***, ***Corner***, ***LB Corner***, ***Left Side***, ***X Evenly*** and ***Y Evenly*** and then click **Clone,** RailClone will generate a new Segment node for each of the selected items.



Adding the Corners

Before we attach the segments to the generator, note that we have four corners but only one segment. In this section we'll demonstrate how you can re-purpose geometry by using the Mirror operator to flip the left bottom segment to create the missing corners.

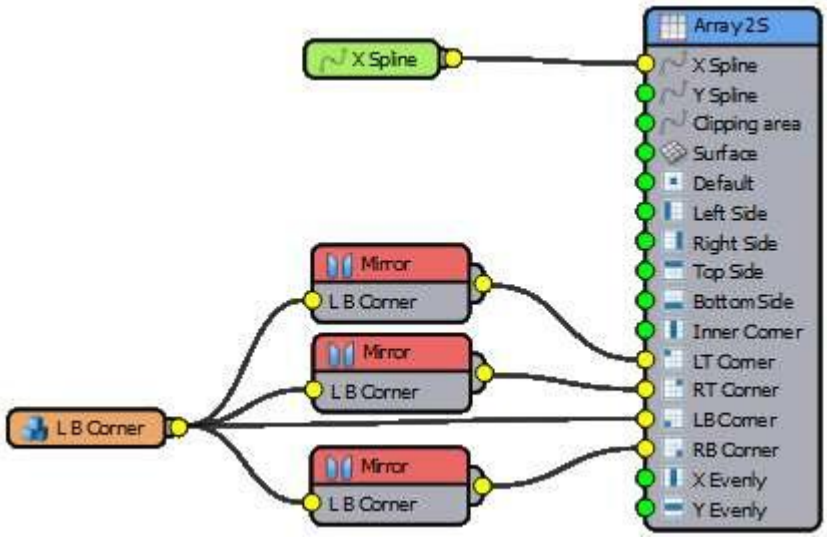
1. Wire the Segment name ***L B Corner*** to the **LB Corner** input of the generator.
2. Create a new **Mirror** operator. Wire the ***L B Corner*** segment to the Mirror operator's input, and then wire the Mirror operator's output to the **RB Corner** input of the generator. The mirror operator has flipped the segment on the X Axis.

3. Next we'll create the top corners. Create a second Mirror operator and open the **Properties** panel. Change the Axis to **Y**.

4. Wire the ***L B Corner*** Segment to the new Mirror operator, then wire the Mirror operator to the **LT Corner** input of the generator. The mirror operator has flipped the segment on the Y Axis.

5. Create a third Mirror operator and open the Properties panel. Change the Axis to **X and Y**.

6. Wire the ***L B Corner*** Segment to the third Mirror operator, then wire the Mirror operator to the **RT Corner** input of the generator. The mirror operator has flipped the segment on the X and Y Axis, completing the corners. At this point your Node tree should look like this



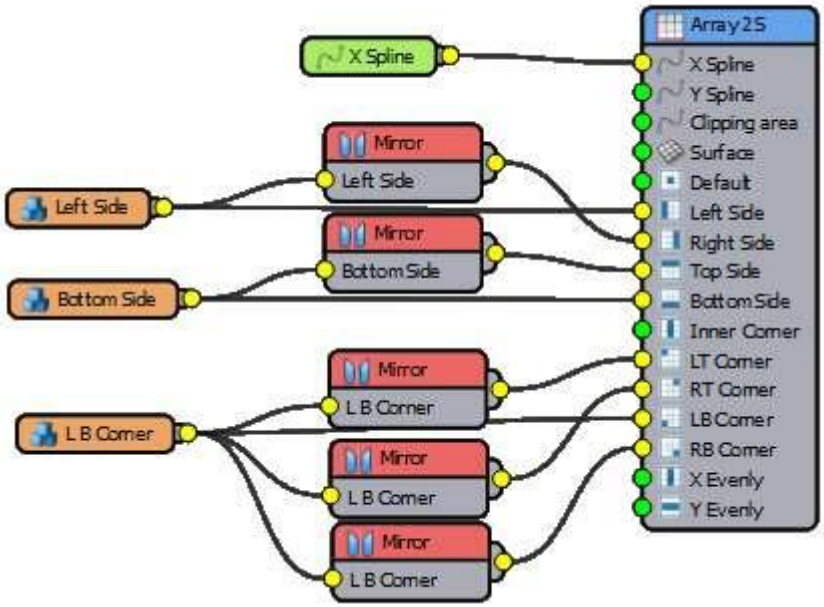
Adding the Sides

Now we'll add a frame to the Top, Bottom left and Right of the glazing. Once again you'll use the mirror operator to re-use segments.

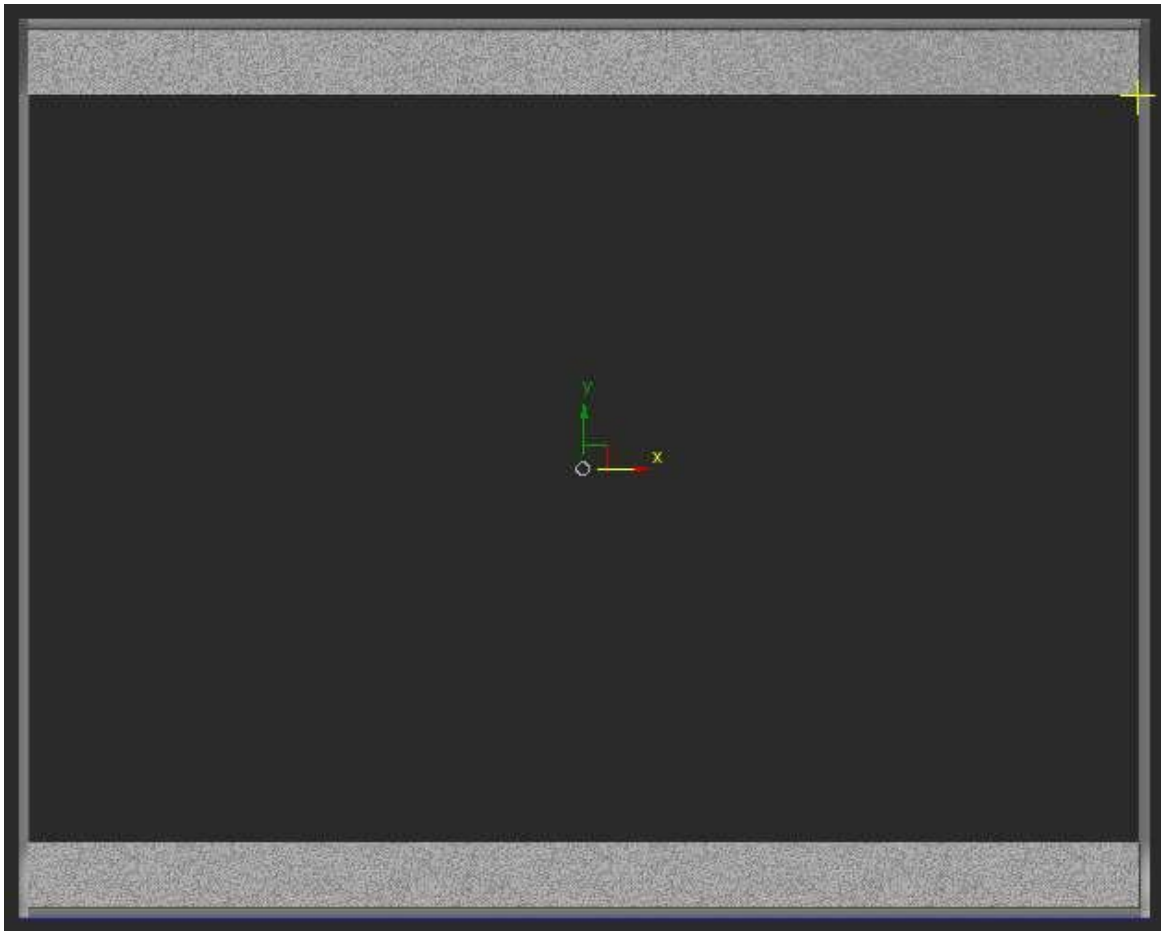
1. Wire the Segment named ***Left*** to the **Left** input of the generator.

2. Create a new **Mirror** operator. Wire the ***Left*** segment to the Mirror operator and then wire the Mirror operator to the **Right** input of the generator.
3. Wire the Segment named ***Bottom*** to the **Bottom** input of the generator.
4. Create another Mirror operator and open the Properties panel. Change the Axis to **Y**.

5. Wire the *Bottom* segment to the new Mirror operator, then wire the Mirror operator to the **Top** input of the generator. At this point your Node tree should look like this



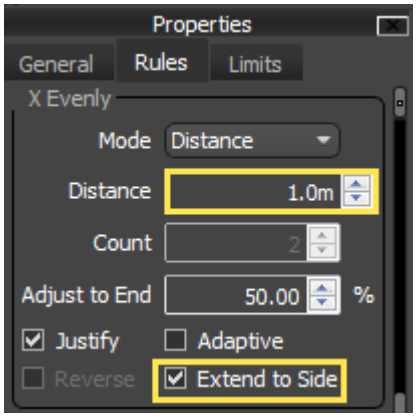
You will have a frame running around the exterior of the array. In the next step we'll start to fill in the centre.



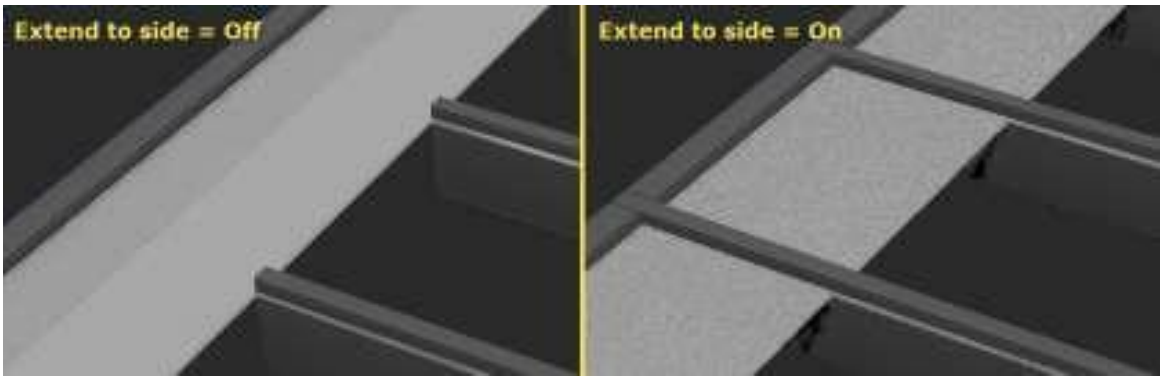
Adding Evenly spaced segment on the X Axis

Segments added to the X Evenly input are regularly positioned along the X axis of the array. Follow next few steps to use this to add vertical separators to the glazing style.

1. Wire the Segment named ***X Evenly*** to the **X Evenly** input of the generator. You'll see regularly space vertical appear in the array.
2. To control the spacing between Evenly segments, select the Generator and go to **Properties** > **Rules** > **X Evenly** and enter a **Distance** value of **1m**.

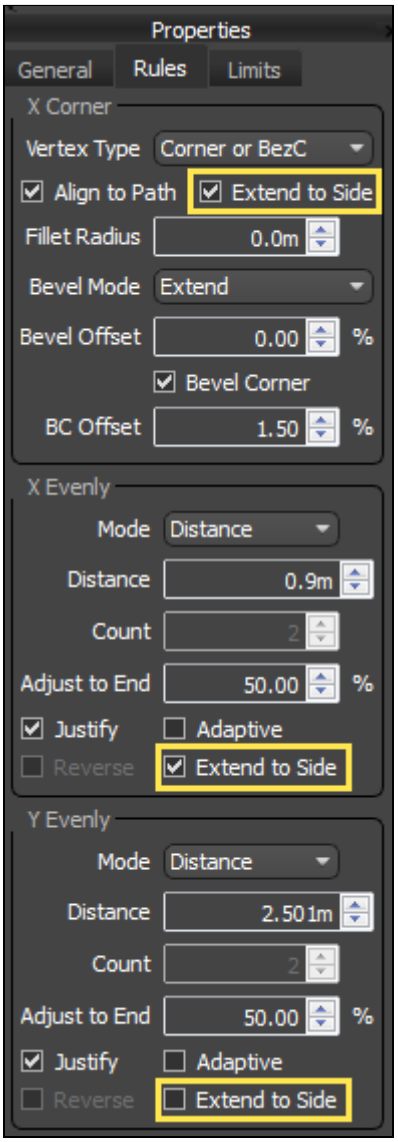


3. Ensure that **Extend to Side** is activated. This will ensure the the Evenly segments go to the edges of the array.



Extend to Side

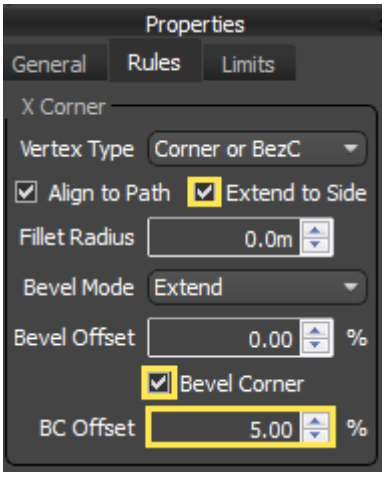
When the **Extend to Side** check-box is enabled for any of the X Evenly, Y Evenly or Inner Corner inputs, The geometry is extended the sides of the array cutting through any segments being used in the Left, Right, Top or Bottom inputs.



Adding Inner Corners

Inner Corners in an A2S generator behave exactly like the Corner input of the L1S Generator. Both of these add geometry only on vertices of a specified type found on the path. By default only Corner or Bezier-Corner vertices are used, but this can be changed in the Generator's properties.To add corners to our Glass Wall style:

1. Wire the Segment named ***Corner*** to the **Corner** input of the generator.
2. Open the Generator's **properties** and ensure that **Rules X > Corner > Extend to Side** is activated. This will ensure the the corner segments go to the edges of the array.
3. Turn on **Bevel Corner** and set the **BC Offset** value to **5%**.



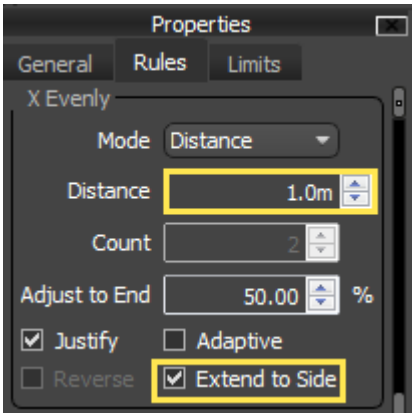
BC Offset

Bevel Corner Offset allows you to adjust the offset of the Corner segments. The distance is measured as a percentage of the length of the corner segment. positive values pull the segment towards the corner, negative values will move the segments away from the corner.

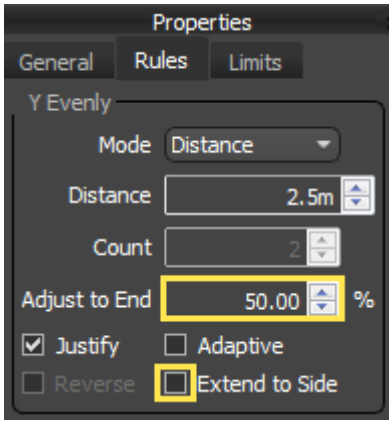
Creating Evenly spaced segments on the Y Axis

Segments added to the Y Evenly input are regularly spaced along the Y axis of the array. Follow the next few steps to use this to add horizontal separators to the glazing style.

1. Wire the Segment named ***Y Evenly*** to the **Y Evenly** input of the generator. You'll see regularly space horizontal bars appear in the array.
2. To control the spacing between Y Evenly segments, go to the Generator's **Properties** and enter a **Y Evenly** > **Distance** value of **2.5m**.



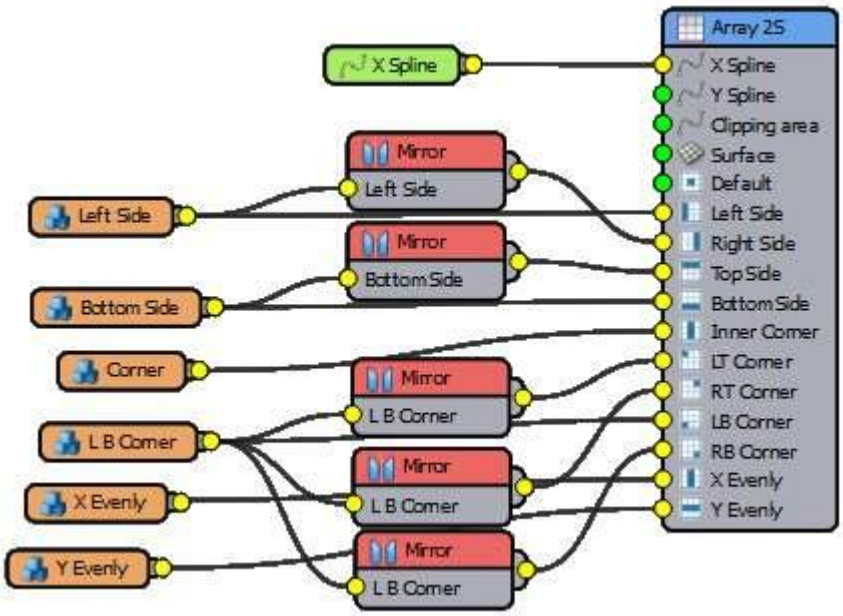
3. Ensure that **Extend to Side** is **unchecked**. In this case we want to prevent the the Y Evenly segments from extending to the sides of the array.



Adding Default segments

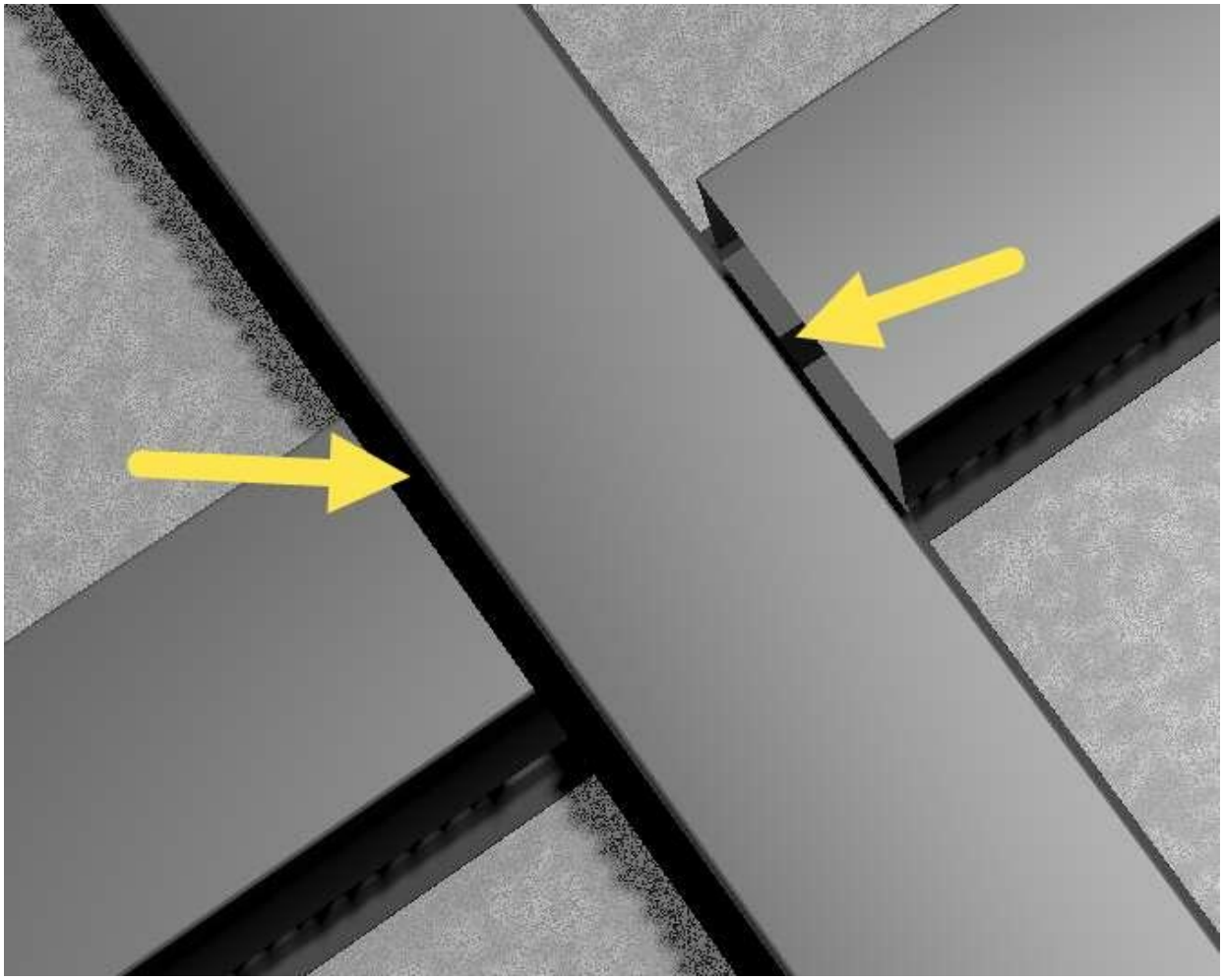
Finally, we'll fill in the remaining gaps with the default segment.

1. Wire the Segment named *Default* to the **Default** input of the generator. You've now used all 12 inputs of the A2S generator to create an adjustable glazing style try experimenting with the Evenly distances to get different results. The final node tree should resemble the image below



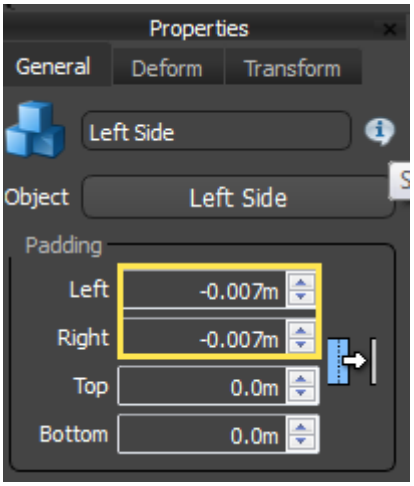
Adjusting Padding

You may have noticed If you look closely at the style you'll that there are some unwanted gaps between the horizontal and vertical elements.



These gaps are caused by the rubber seal which extends beyond the metal frame. To correct this we need to add some padding values to the Vertical elements. For a more detailed explanation of padding, please see the chapter called How to align, overlap and transform geometry

1. Select the ***Left*** Segment and open the Properties. Enter a value of **-0.007m** for the **Left Padding** and **Right Padding** parameters.

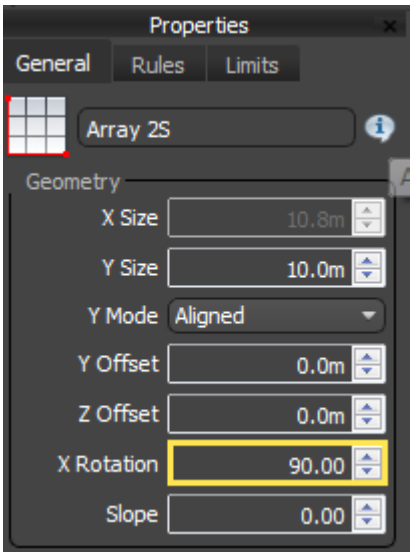


2. Repeat this procedure for the ***X Evenly***, ***L B Corner*** and ***Corner*** segments. The gaps will now be removed.

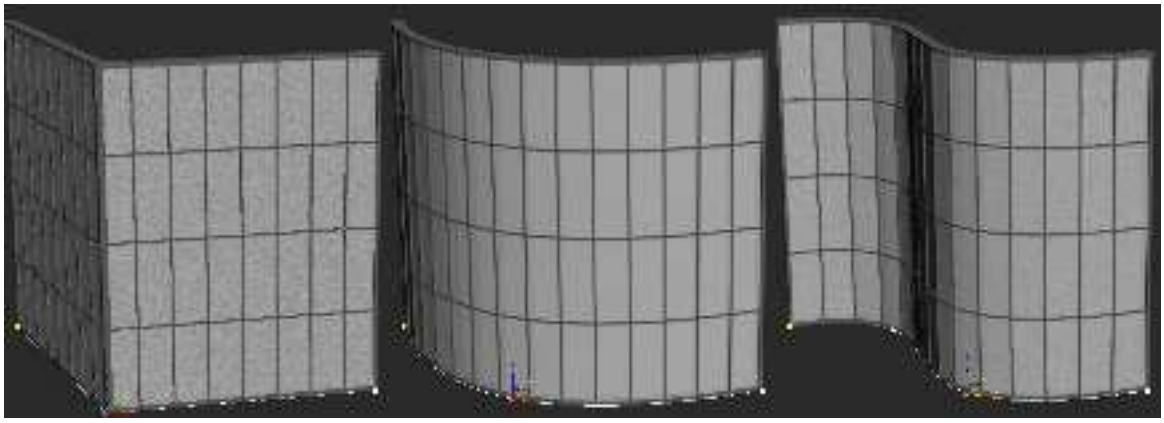
Rotate the Array

This completes the style though it is currently flat on the ground plane. For a glazing style we need to rotate the entire mesh so that it is vertical. To do this:

1. Select the **Generator** and open the **Properties**.
2. Enter a value of **90** degrees for the **General** > **X Rotation** parameter.



3. You can now try reshaping the spline to get different shapes of glazed wall.



9.3 Related Tutorials

create_a_curtain_wall_building

Create a curtain wall building

A further example of using the A2S array to create a curtain wall. In this example we do things slightly differently and use a Spline for both axes of the array, making it possible to create double-curved façades.

Stadium Tutorial Part 5

In the final part of the stadium tutorial we demonstrate how to use an A2S generator to create the roof. You'll learn how to create the style and also export attributes so that you can adjust the depth of the roof and spacing of the supports from the Modify panel

Create a seaside promenade

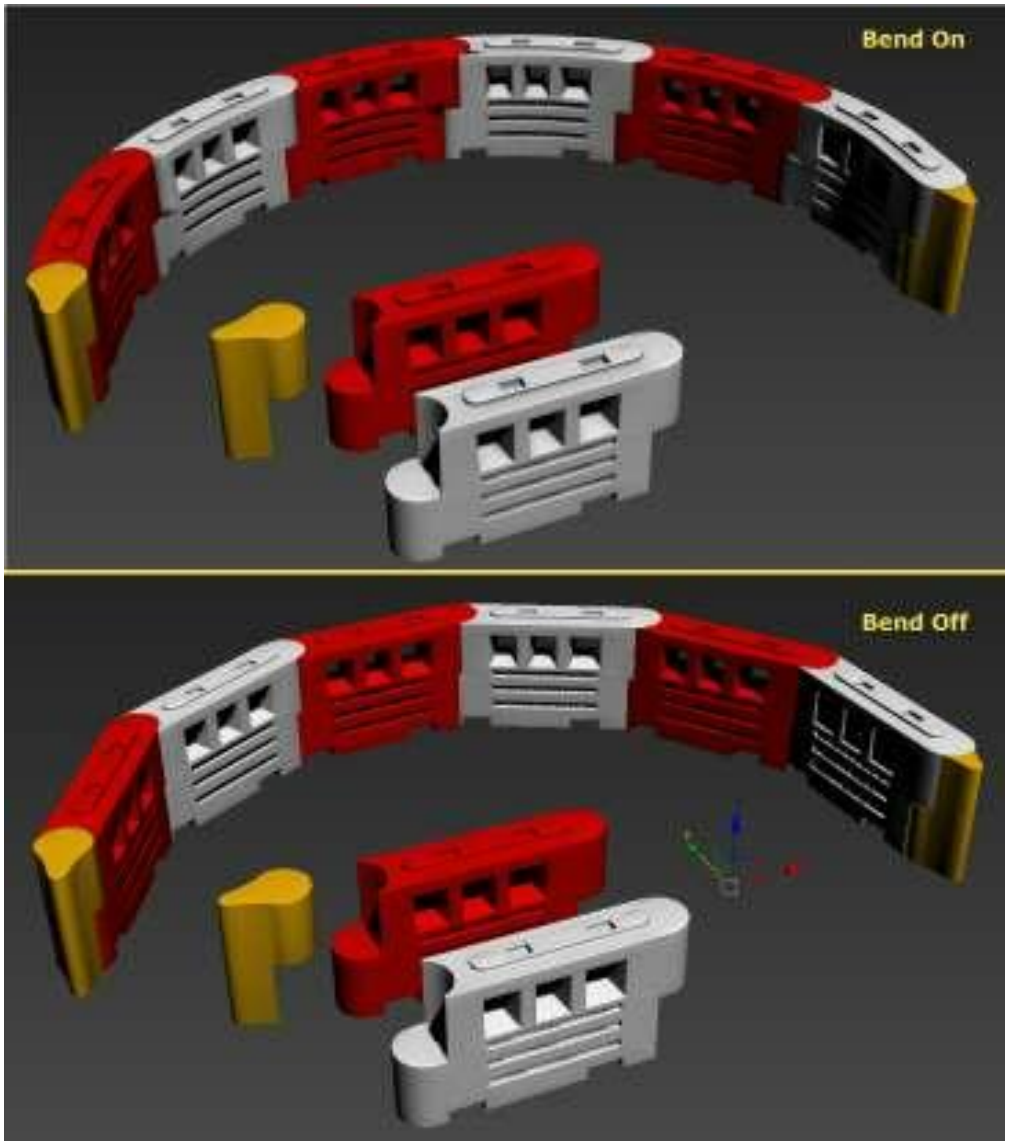
In the last 2 parts of this tutorial we explain how to use the A2s generator to create a paving style and a huge sea wall using RailClone's formidable instancing power to create the geometry from individual bricks!

Documentation

For more information please see the 2D arrays - Generator A2S section of the online documentation.

10 Techniques to deform geometry

One of the most powerful features of RailClone is the ability to deform geometry to follow a path. Segments can be set to follow the curvature of a spline on the X and Y axes, with a host of additional settings to control how geometry is deformed on the Z axis. In this chapter we'll explore the many modes available, starting with turning Bend **Off.** Deforming geometry isn't always desirable, sometimes you want to ensure it retains its original shape. Take the traffic Barrier style illustrated below for example. In the top image the segment has bend turned on, as a result each segment is smoothly deformed to follow the base spline. It looks good, but this isn't how these barriers work in real-life. Instead we need the barrier segments to follow the spline without deforming, as in the bottom image.

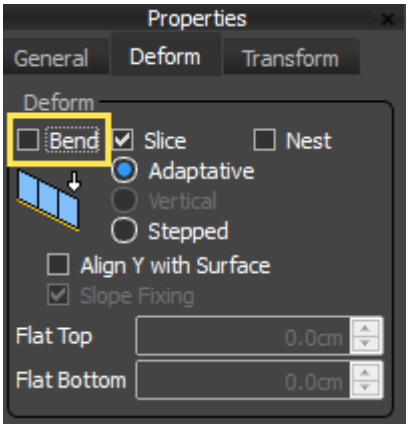


10.1 Exercise: Turning off Bend

New Segments have the bend option enabled by default. In the majority of cases this is the correct behaviour, but to demonstrate how to turn Bend off, In this exercise you'll correct the barrier style used in the illustration above. Open the file called bend_slice.max and follow these steps:

1. Select ***rc_barrier_plastic***, go to the **Style** rollout and click 

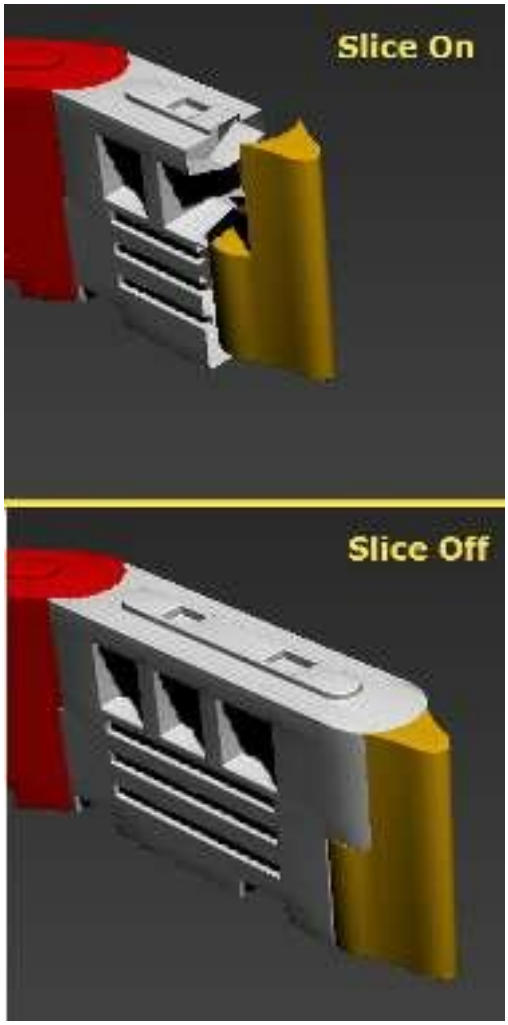
   to open the **Style Editor**.
2. For each of the 3 Segments, go to **Properties** > **Deform** > **Deform** and turn off **Bend.**



3. The Barrier will now follow the path correctly.

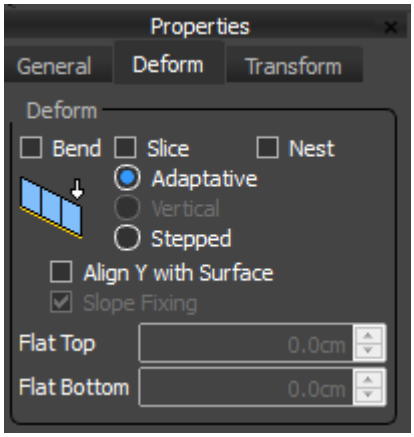
10.2 Exercise: Turning off Slice.

Next to Bend in the interface, there's an option to Slice the segment. When this is turned off, RailClone cuts off the part of the last segment that falls outside of the path. In most cases this is the correct behaviour, but occasionally you'll want to turn this option off too. If you look at the the barrier style again, you'll see that the final Segment in the array is being sliced. In this instance, turning slice off gives us a much better result, though it's important to note that this does result in the segment extending beyond the end of the spline.



To turn off Slice:

1. Make sure that ***rc_barrier_plastic*** is selected with the **Style Editor** is open.
2. For each of the 3 Segments, go to **Properties > Deform > Deform** and turn off **Slice.**



3. The Barrier style is now correct.

What to do if deformation is not smooth

If the deformation of segments is jagged, this is often because the interpolation value of the spline being used as a base object is too low. Instead of editing the spline, RailClone provides a feature to improve smoothing by automatically adding additional interpolation steps. To use this feature open the **Style** rollout and increase the **Curve Steps** value until you get smoother results.

Optimising

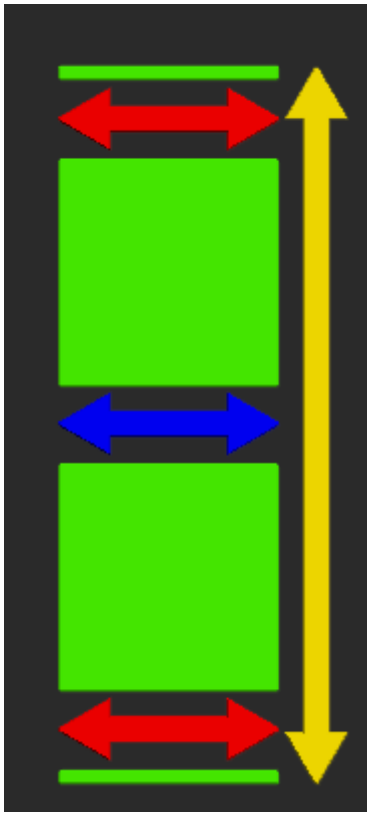
Segments which have been deformed or sliced are no longer instances and contribute to the overall polygon count. To create optimal renders disable Bend and Slice for all segments that don't need it.

10.3 Deforming on the Z Axis

RailClone provides 3 additional options to to control how segments deform on the Z axis: Adaptive, Vertical and Stepped. To understand these modes in more detail, try the following exercise:

Exercise: Testing X Deform modes

To illustrate how the modes work, open the file named z_deform_modes.max. This file contains a simplified style with a single segment. The geometry, shown below, has been designed to make it clear how the 3 different modes affect the deformation.



The **yellow** arrow helps to illustrate how the current deformation mode affects the **vertical** elements of the geometry, the **red** arrows illustrate how the current deformation mode affects the horizontal elements at the **top** and **bottom** of the geometry, and the **Blue** arrow illustrate how the deformation mode affects the horizontal elements in the **middle** of the geometry.

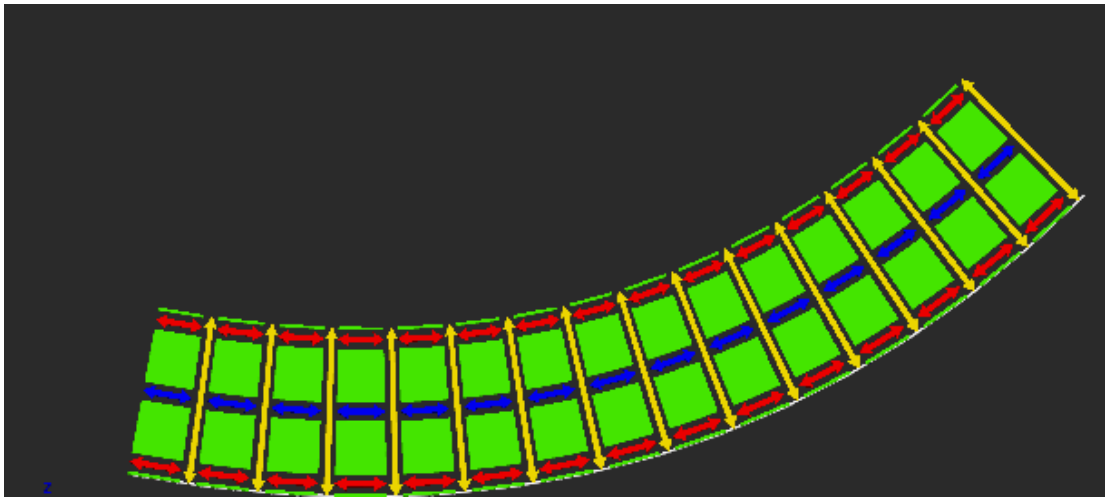
1. To test the modes. select the RailClone object called *rc_z_modes*, go to the **Style** rollout and click

   to open the **Style Editor**.
2. Select the segment and and go **Properties** > **Deform** > **Deform**.
3. Follow the steps below to activate each of the modes. After each one, scrub the timeline. The base spline is animated so you can see how the deformation reacts to the changing shape of the path.

Adaptative

In this mode , the segment's verticals (yellow arrows) are aligned perpendicular to the path; while the horizontal elements (red and blue arrows) follow the path.

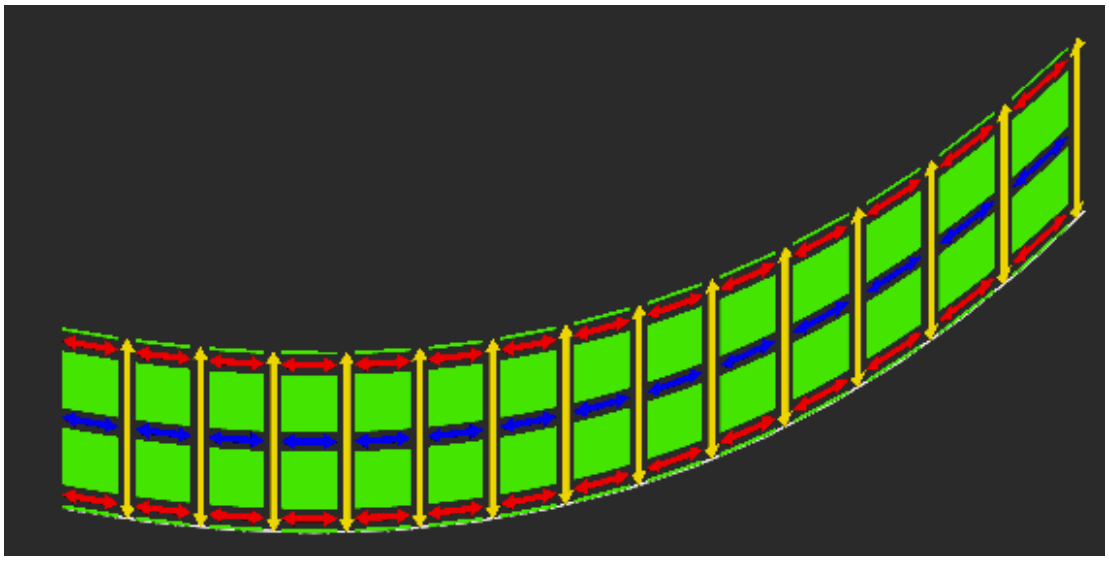


To enable Adaptive deformation mode:

1. Change the **Deform** mode to **Adaptive**.

Vertical

In this mode, the vertical elements (yellow arrows) remain upright, but the horizontal elements (blue and red arrows) are deformed on the Z axis to match the elevation of the path.

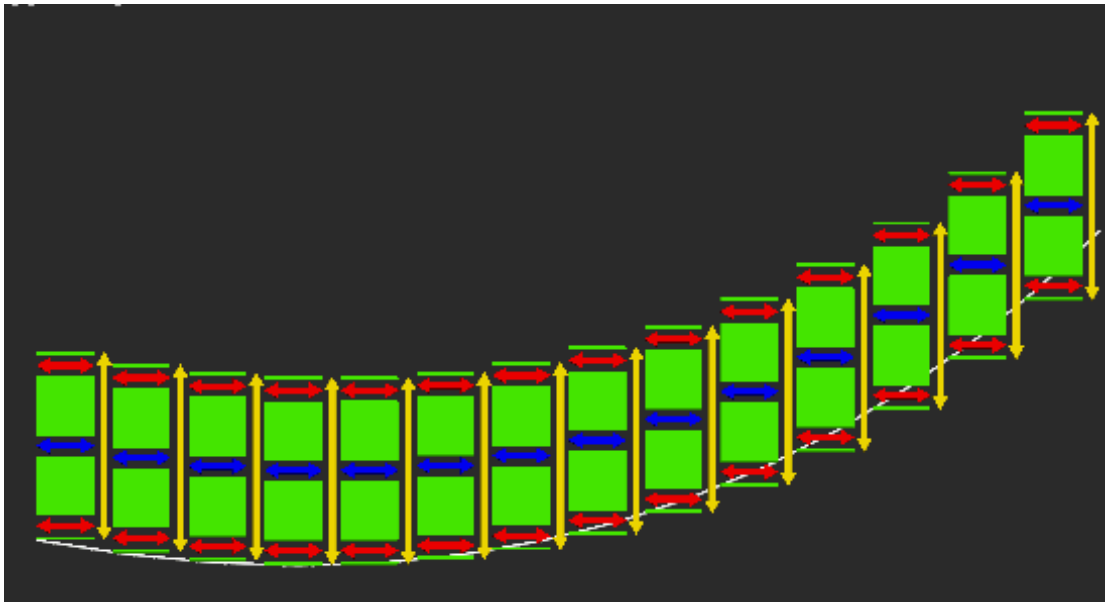


To enable Vertical deformation mode:

1. Change the **Deform** mode to **Vertical**.

Stepped

In this mode, no deformations are applied in the Z axis, and the segment preserves its vertical alignment.



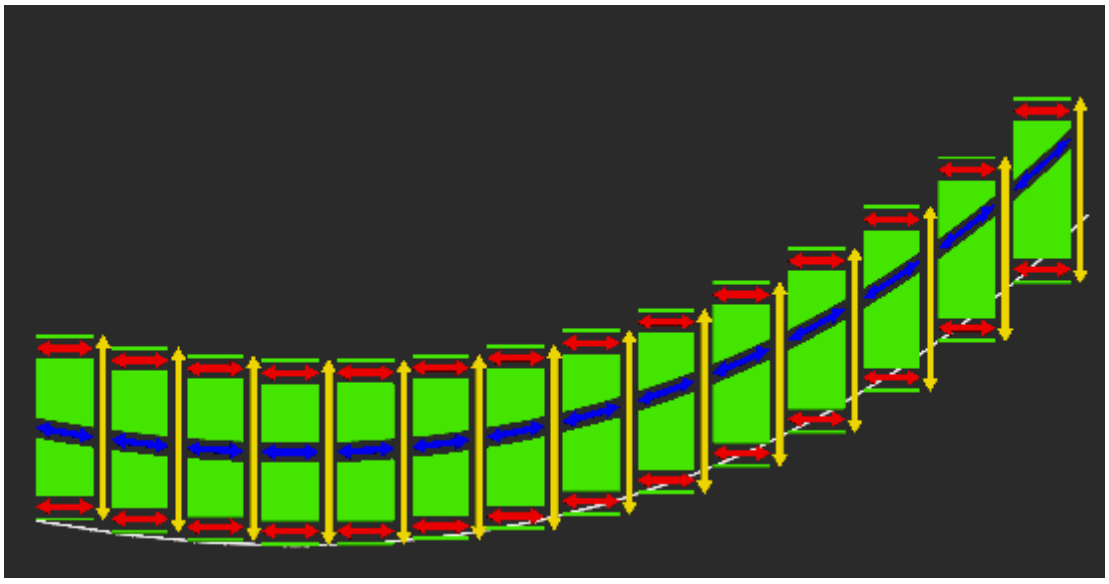
To enable Stepped deformation mode:

1. Change the **Deform** mode to Stepped.

In addition to these 3 modes, there are 2 additional **hybrid** modes that combine the Vertical and Stepped deformation algorithms.

Vertical mode with Flat Top/Flat Bottom

In this mode the top and/or the bottom of a geometry is stepped (red arrows) while the areas in between follows the spline (blue arrows). The Top and Bottom values define the distance (on the Z axis) that the effect is applied, measured from the top or the bottom of the geometry.

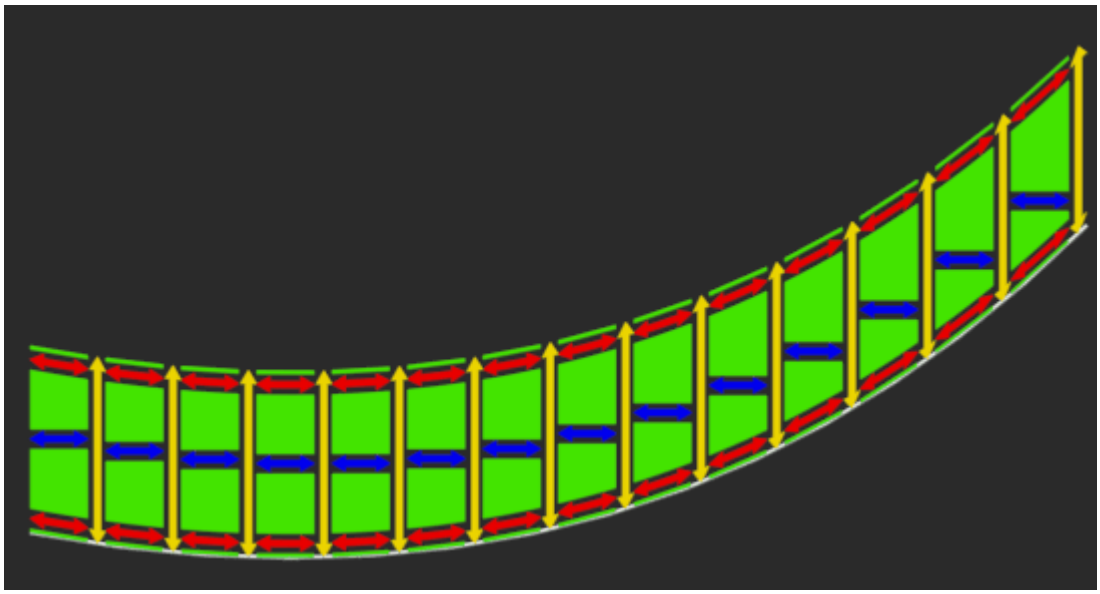


To enable Flat Top or Flat Bottom deformation mode:

1. Change the **Deform** mode to **Vertical**.
2. Enter a distance in for the **Flat Bottom** value and/or enter a distance for the **Flat Top** value. In the example scene, a value of 0.8m works well.

stepped mode with Vertical Top/Bottom

In this mode the top or bottom of a segment (red arrows) deform and follow the spline, leaving the middle area (blue arrows) stepped. The Top and Bottom values define the distance (on the Z axis) that the effect is applied, measured from the top or the bottom of the geometry.



To enable Vertical Top or Vertical Bottom deformation mode:

1. Change the **Deform** mode to **Vertical**.
2. Enter a distance in for the **Vertical Bottom** value and/or enter a distance for the **Vertical Top** value.

10.4 Related Tutorials

Creating Stairs

This tutorial looks at the different ways segments can be deformed on the Z-Axis. After explaining the deformation modes we explore how to put them to good use to create a straight and spiral staircase.

Documentation

For more information please see the Segments section of the online documentation.

11 How to align, overlap and transform geometry

In this section we'll look at how to adjust the padding, alignment, and transform settings of a segment. This can be useful where you wish to reuse a segment multiple times but with different properties, add randomisation to the transforms, or adjust the spacing between adjacent segments.

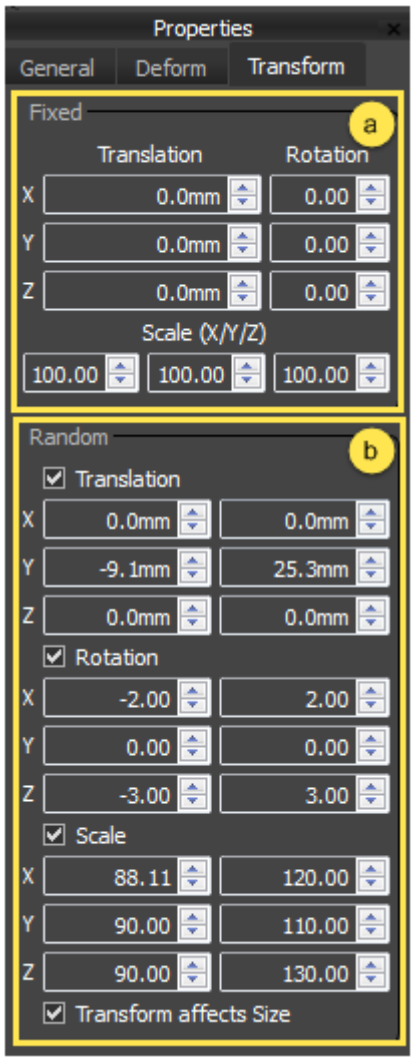
11.1 Transforming Segments

Fixed Transforms

Once a segment is added to RailClone it's possible edit its rotation, translation and scale values from the Properties Panel. To access these options:

1. Select a Segment node (
   

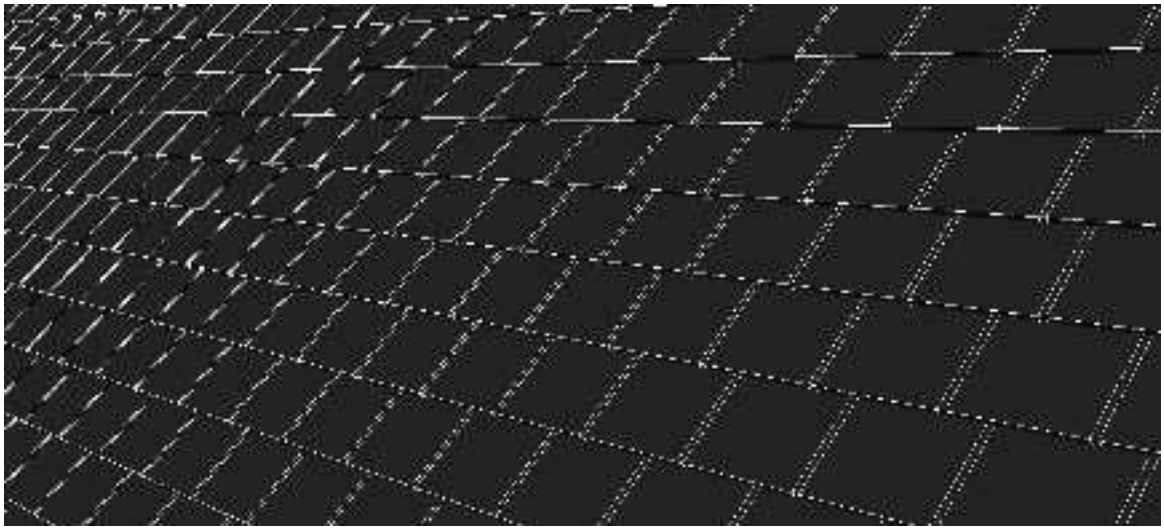
).
2. Go to the **Properties Panel**.
3. Go to the **Transform** tab. In the Fixed group (a) you'll see there are parameters that enable you to translate, rotate and scale the segment's geometry. Underneath this, the Random group (b) allows you to randomising these properties with in a range. You can try out these parameters by following the exercise in the next section.



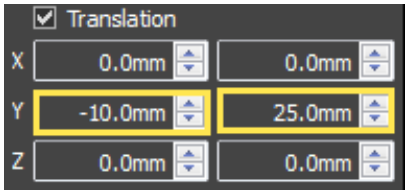
Exercise: Randomising Transforms

In this section you'll learn how to randomise the position, rotation and scale of segments using roof slates as an example. To follow this procedure, download and open slate_tiles.max.

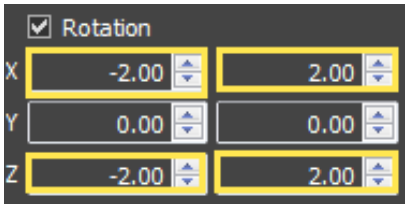
1. With slate_tiles.max open, select the RailClone object called ***rc_slates*.** The slate geometry is quite high poly so In the viewports this RC object is using **Box** display mode to help maintain viewport speed. As you can see the slate pattern currently has no variation.



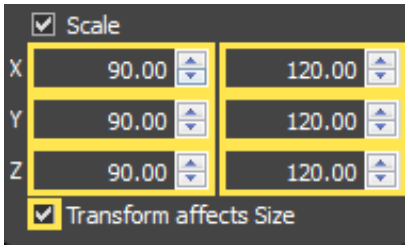
2. we can start adding some irregularity to this by randomising the tile's position. Open the **Style Editor** and select the Segment node called ***Slate_002***. Go to the **Properties panel** select the **Transform** tab and turn on **Translation** in the **Random** group. Each axis can be randomised independently by entering a **Minimum** and **Maximum** value, measured in scene units. In this example we'll enter a value of **-10mm** for the **Translation > Y Min** value, and a **Y Max** value of **25mm.**



3. To randomise Rotation, activate **Transform > Random > Rotation** and enter **Minimum** value of -2 and **Maximum** value of 2 degrees the **X** and **Y** axes.



4. To randomise Scale, activate **Transform > Random > Scale** and enter **Minimum** value of **90** and a **Maximum** value of **120** for the **X**,**Y** and **Z** axes.

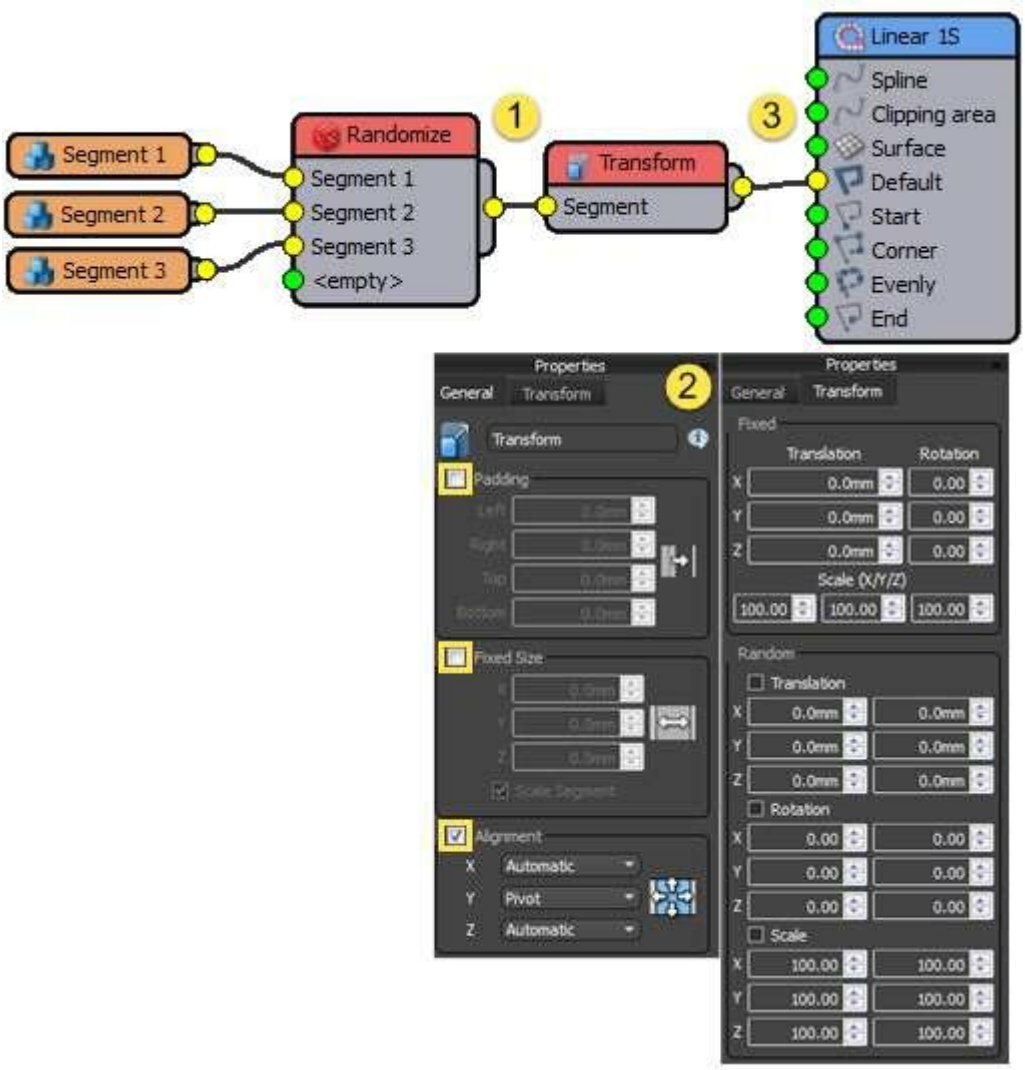


5. For this style these changes in transform settings should affect the position of adjacent segments. To do this make sure that **Transform Affects Size** is left **On**. If you would like to use the size of the original segment before transforms, turn this setting off. The image below illustrates the render before and after randomisation.

Top: no transform randomisation | Bottom: with transform randomisation (exaggerated)

The Transform Operator

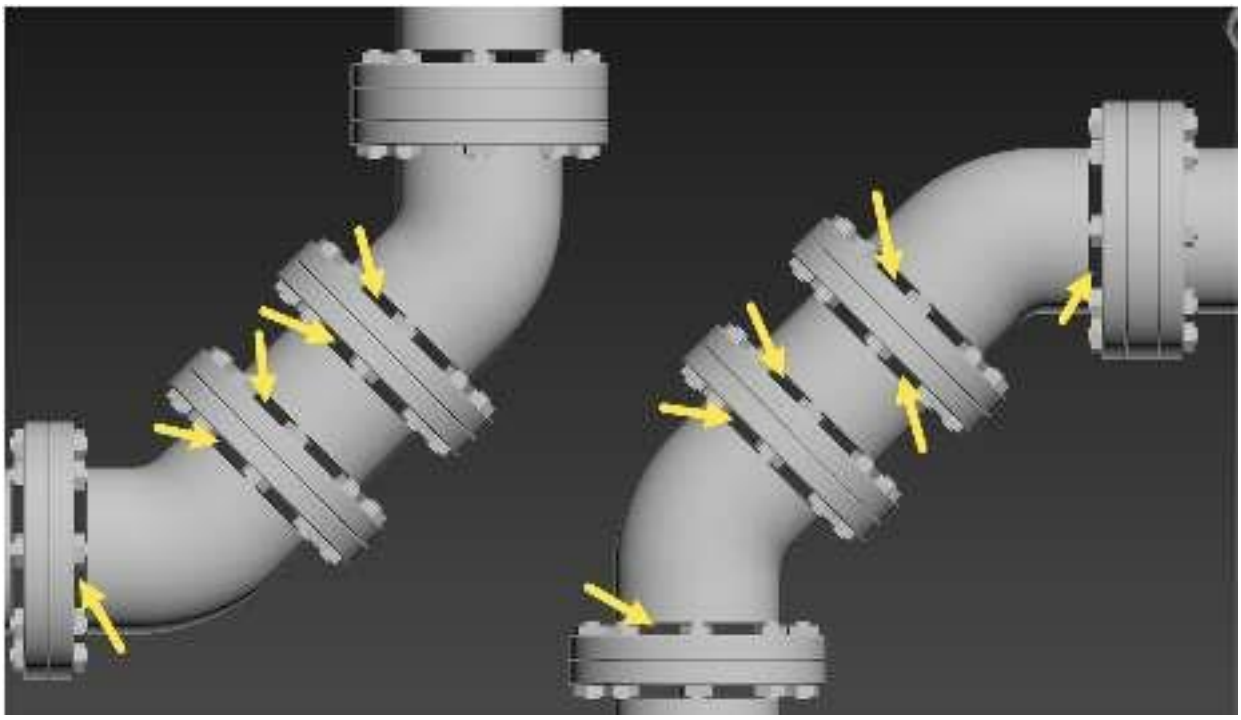
Instead of adjusting the transform values from the Segment node's properties, it is also possible to use a Transform operator to access the same settings.This can be useful if you want to change multiple segments that are attached to a Sequence, Compose, Randomise, Selector, or Conditional operator. Using the Transform operator can save you from having to enter the same information multiple times and makes updating your styles much quicker and easier. The basic procedure for using the Transform operator is as follows:



1. Wire a node to the Transform operator's **input**.
2. In the Transform node's properties, activate any settings you wish to use by clicking on the appropriate check-box. This will override the values set in the Segment's own properties.
3. Connect the Transform operator's **output** to a generator or another operator's **input**.

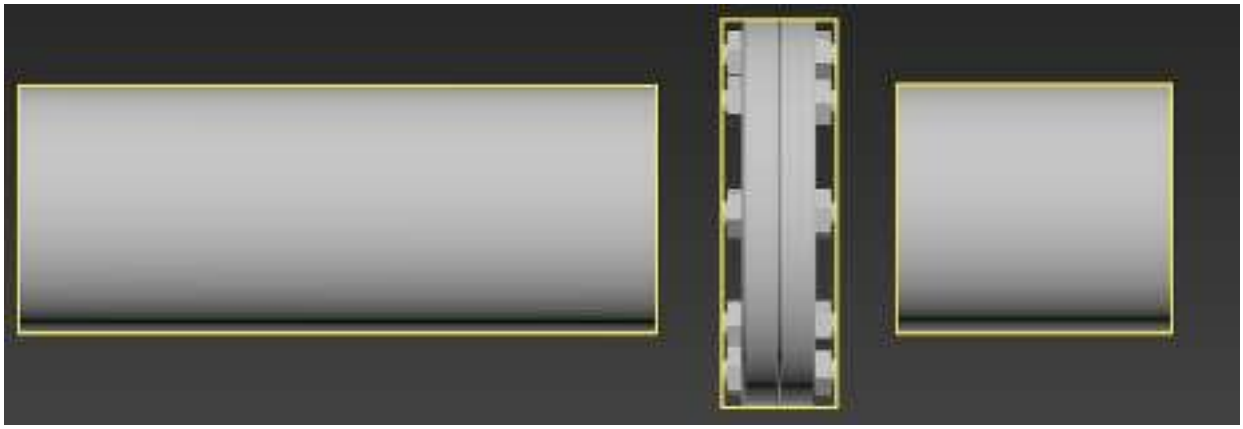
11.2 Segment Padding

Each segment has a length and height extracted from the bounding box of the source geometry. This dimension is used to control how the segment will be positioned alongside others in the array. For some geometry this may produce undesirable results, as the bounding box will always be determined by the largest part of the mesh. For example, load the segment_padding.max exercise file. In this scene is a relatively simple pipe preset that uses bolted collars for the corner and evenly inputs. If you look closely at the style you'll notice there are some unwanted gaps:



Unwanted gaps in the Pipe style

This is cause by the bolts, as they define the size of the bounding box, pushing the pipes away from the connector and causing a gap:

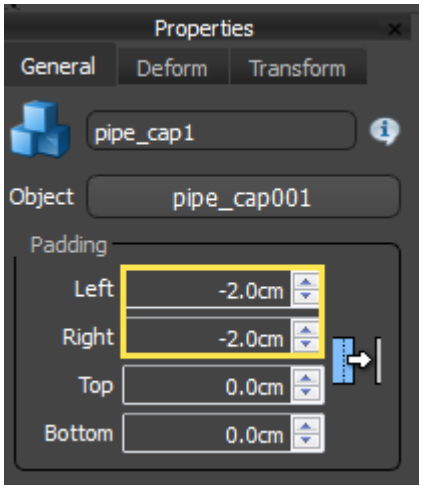


The 3 segments used in the Pipe style and their bounding boxes (in yellow). Note how the bolts will prevent the pipes from properly connecting to the collar.

Fortunately RailClone includes a **Padding** parameter to compensate for this kind of geometry. This feature allows you to modify the default size, increasing or decreasing the distance between the segment and others to the left, right, top and bottom sides. To see how it works, let's prevent some leaks and correct this style so that the pipes meets the collar correctly.

Exercise: Editing Segment Padding.

1. Open segment_padding.max and select *rc_pipes*.
2. Open the **Style Editor** and select the **Segment** node named ***pipe_cap1***.
3. Go to the **Properties** Panel and enter a value of **-2cm** for the **Left** and **Right** padding. The gap between the collar and the pipe will close up.



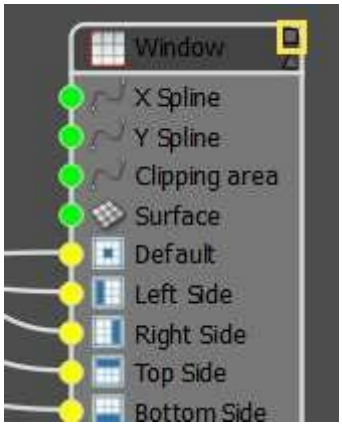
11.3 Generator Padding

Generators can also use padding values to offset the entire array. These are used to add offsets to the **left**, **right**, **top** and **bottom** of A2S generators, or the **start** and **end** of L1S generators. All calculations, such as evenly spacing, are based on the new lengths of the array **after** the offset has been applied. This is particularly useful for situations where you need multiple generators to jigsaw together, as we'll see in the following example of a window inside a surround.

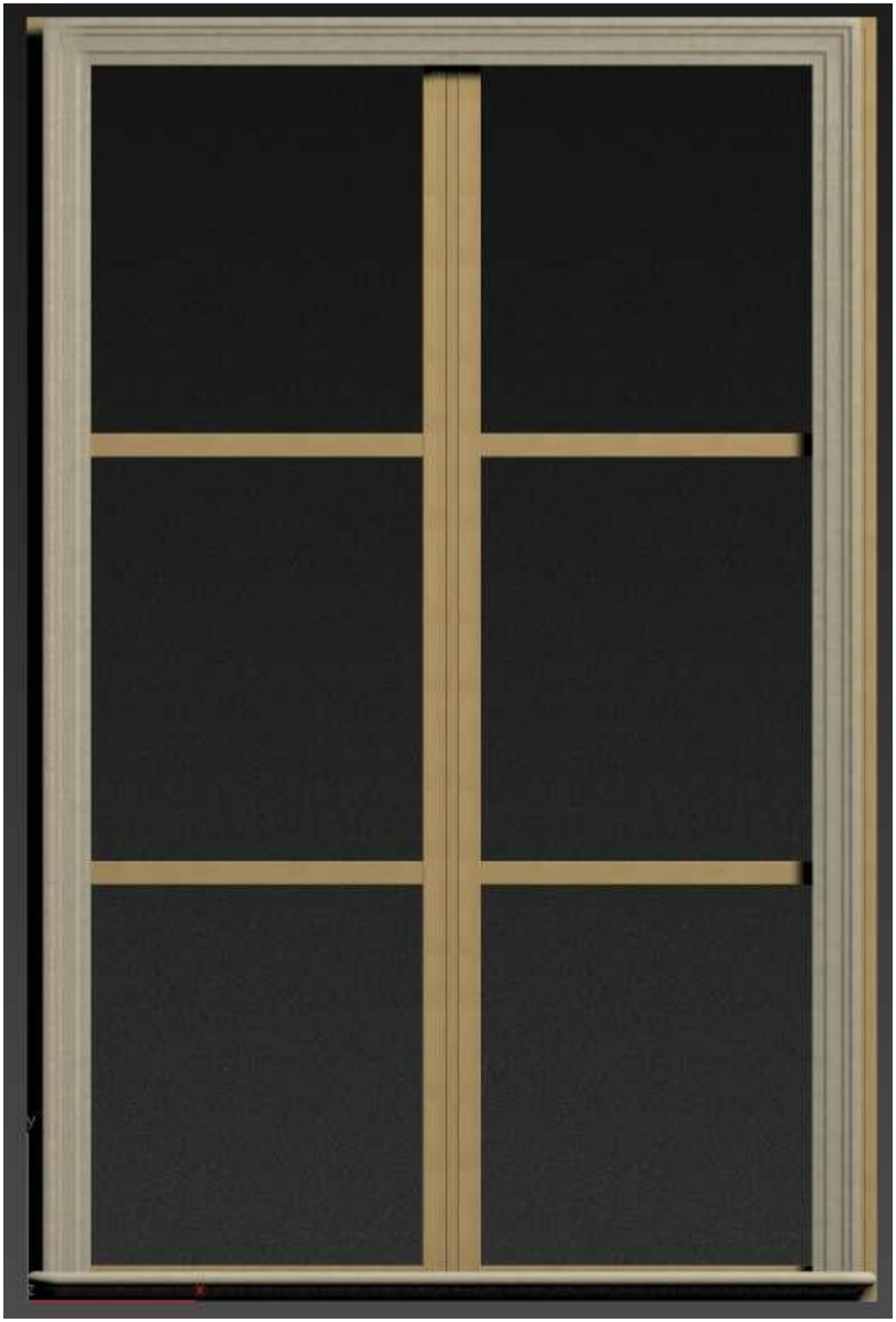
Exercise: Editing Generator Padding

1. Open the file named generator_padding.max.
2. Select the object called ***rc_window***. At the moment only the surround is displayed.

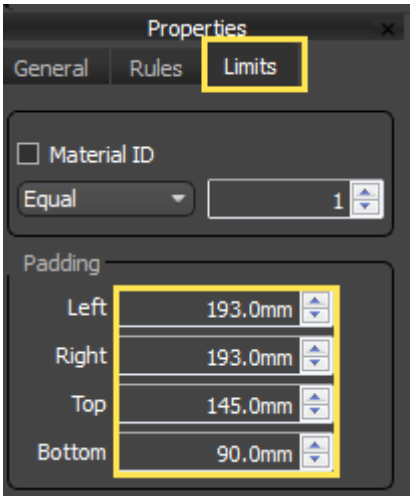
3. To see the window, open the style editor and and find the generator called ***Window.*** This generator has been disabled. To enable it, click on the check-box found in generator's top right hand corner.



4. This style has one generator to create the surround and another for the window, Both generators use the same length values to determine their size on the X and Y Axes. Because of this the window on the frame overlap.



5. To prevent the window from overlapping the outer surround, padding is added to the top, bottom, left and right of the generator. To do this click on the **Window** generator and go to the **Properties Panel**. Select the **Limits** tab and enter a value of **193mm** for the **Left** and **Right** padding, **145mm** for the **Top** padding, and **90mm** for the **Bottom** padding.



6. The window now sits neatly inside the surround. Close the Style Editor and adjust the **size** from the **Parameters** rollout, you'll notice that the offset remains constant irrespective of the size of the surround. Also of note, this style uses Evenly Count mode to ensure that the amount of vertical and horizontal elements remains the same as the window is resized.

11.4 Related Tutorials

Random Books Tutorial

This tutorial explains how to use the Random Transform feature to create books of different sizes and introduce some chaos to their distribution. In this tutorial you will also learn how to use the Randomise operator to randomly select geometry from a number of segments.

Masonry Wall

The Masonry Wall tutorial makes good use of padding values to compensate for the overhang of the cap to the top of the pillars. This tutorial is intended for new users of RailClone, and provides a thorough introduction to some of the most commonly used features. By completing this exercise you will be able to use the built in library, understand the style editor, and easily create new styles.

Parquet Floor Tutorial

This tutorial explains how to model parquet floors from individual planks. The 2D array generator is used throughout and provides an excellent way to create a wide variety of floor patterns using only a single enclosed spline to define the boundary. The planks are made to fit together correctly and make 3 different patterns by editing their padding values. Also of note is the Floor library that ships with RailClone. You can study these styles to learn more ways to create floors.

Stadium Tutorial Part 2,3, and 4

The stadium tutorial uses a number of L1S arrays to create advertising hoarding, seating and the crowd. Padding, alignment and transform randomisation is used throughout to create the seats, crowds, and flags.

Create a seaside promenade

Parts 2 and 3 of this tutorial use Padding values extensively to ensure that the segments fit together correctly.

Documentation

For more information please see the Segments, 2D arrays - Generator A2Sand 1D arrays - Generator L1S sections of the online documentation.

12 How to combine, select and sequence geometry

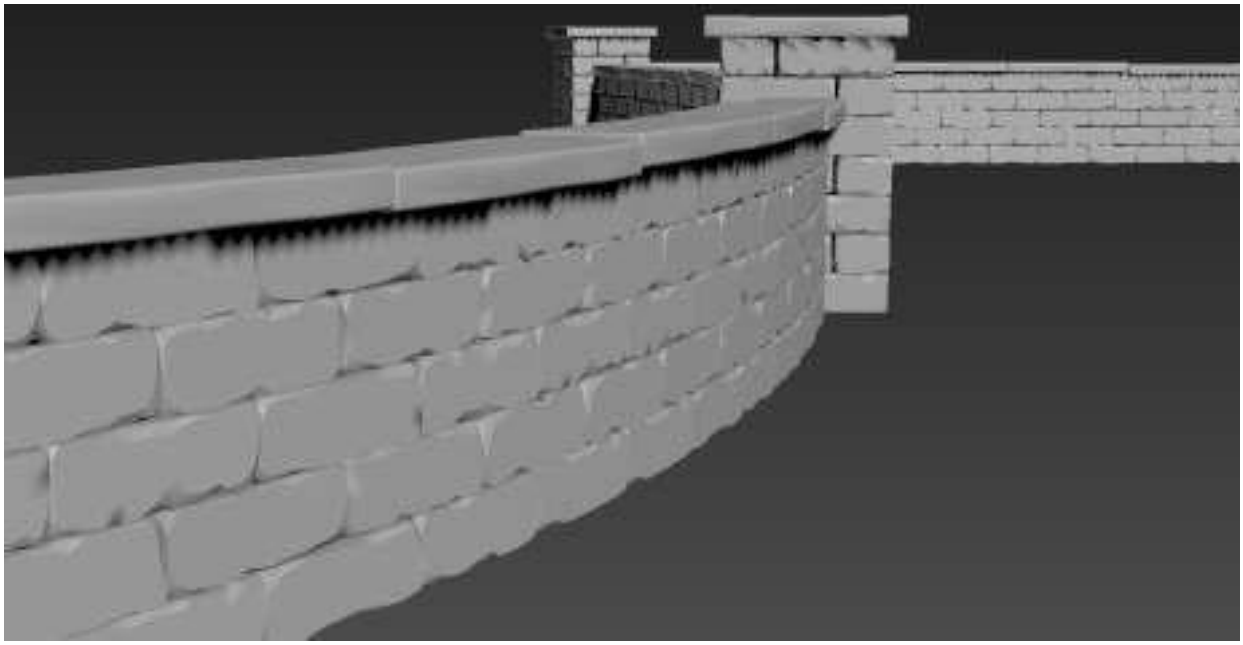
Once segments have been added to a style, operators allow you to apply all kinds of modifications. For example you can use them to mirror a segment, combine multiple segments to create a new composite object, create patterns, randomise geometry or material IDs, and much more. In this introductory guide we will focus on operators that enable you to sequence, select, and combine segments, if you would like to see the full range of operators, including instruction for use, check out the full documentation.

12.1 Operators that select segments

Let's start by looking at two operators that allow you to select from a number of segments. The most literal version of this is the **Selector** operator. This allows you to choose from a list of segments using a numerical index. This index can be adjusted manually from inside operator's properties or it can be wired to a numerical node to be edited from the modify panel without requiring you to open the style editor. The operator also has an alternate mode the derives the index number from the material ID assigned to a spline segment.

Secondly we'll look at the Randomise operator, this returns a random Segment from the list of inputs. Each segment has a presence value that determines the probability of it being selected.

To explore this and the next few operators let's create a brick wall style. The final geometry looks like this:



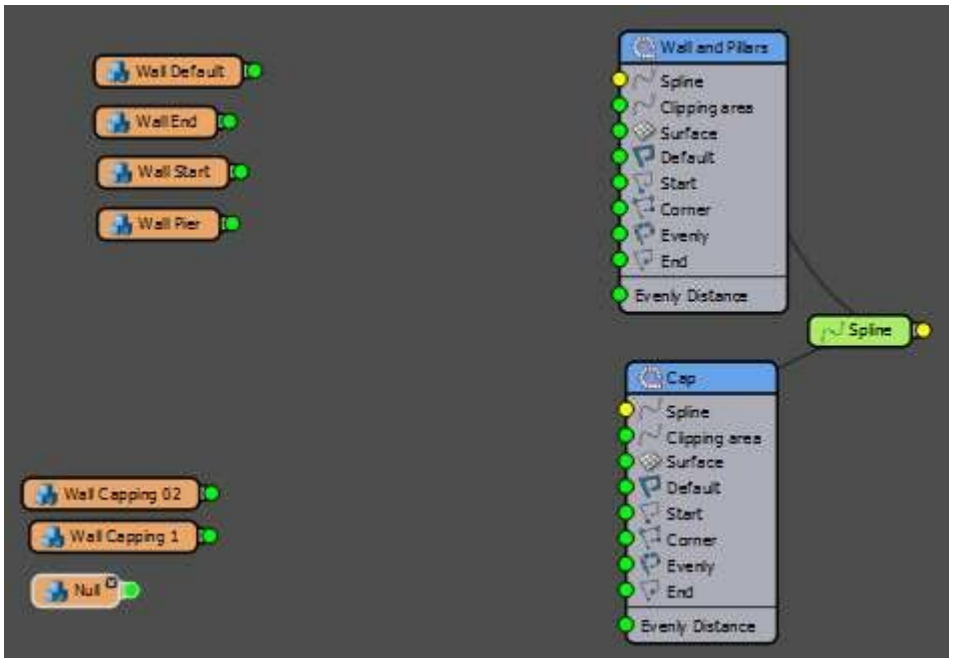
This style is constructed using these segments:

To get started, open the file name *operators_1.max*. To make things easier there's already a RailClone object in the scene and all the segments have been added for you. To ensure the pieces fit together correctly, padding values have been added to the Wall Default, Wall Start, and Wall Pier segments. If you need to review padding, please see Chapter 11. To add operators and create the rest of the style, follow these steps:

Exercise: Create and randomise the wall

1. Select the RailClone object called ***rc_wall***.
2. Go to the **Style** rollout and click
   

to open the **Style Editor**. As well as all the segments you'll find two generators that share a spline. The top generator is used to create the main wall and the second generator adds the capping. As mentioned previously, it is often easier to break the style across multiple generators.

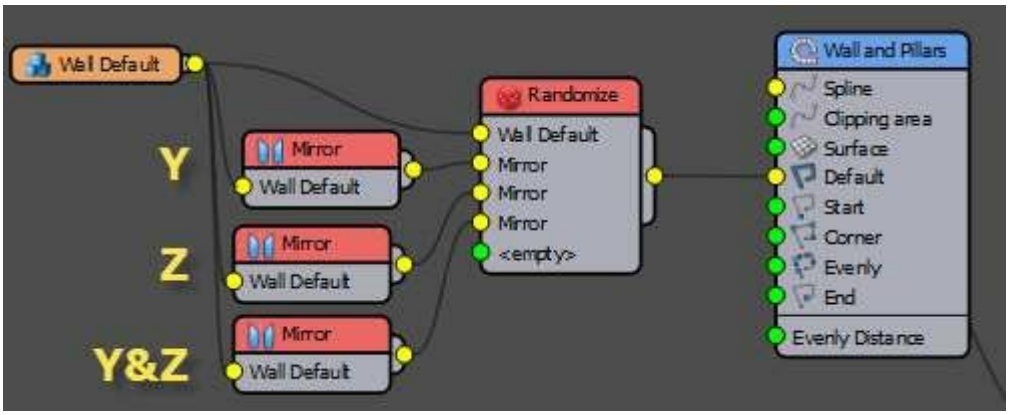


3. Let's start with the wall, create a new **Randomise** operator and wire it to the ***Wall and Pillars*** Generator's default input. We'll use this to add some variation to the wall geometry.
4. Wire the segment named ***Wall Default*** to the **Randomise** operator's first input.
5. There isn't actually any additional geometry for the wall's default input, so we'll create variation by using mirrored clones of the existing brick segment. To do this, create a new **Mirror** operator by dragging from the **Items list** to the **Construction View**.

> ### Mirror Operator
>
> This operator makes a mirror copy of a segment's geometry using any combination of the X,Y, or Z axes as a mirror plane.

6. Wire the ***Wall Default*** segment to the **Mirror** operator's **input**, then wire the Mirror operator to the **second input** of the **Randomise** operator.
7. Click on the **Mirror** operator, go to the **Properties** panel and change the Mirror plane from X to **Y**.

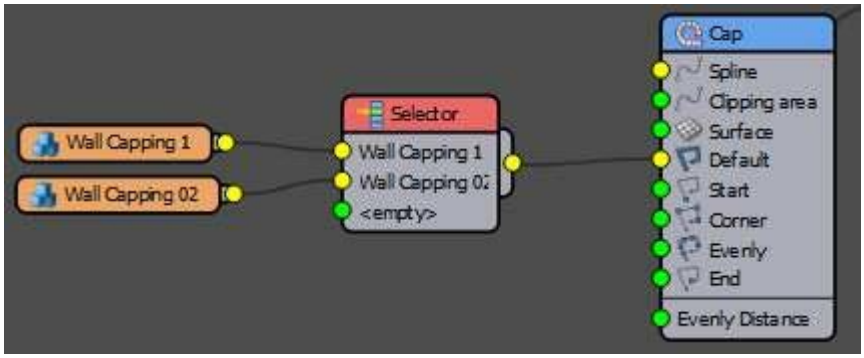
8. We can now Copy and Paste this operator to create 2 more variations. To do this, right click on the node and select **Copy**, then right click again and select **Paste**.
9. Select the new Mirror operator, Change the mirror plane from Y to **Z** and wire the output to the third input of the **Randomise** operator.
10. Copy and paste the Mirror operator a final time and change the axis to **Y and Z**. Wire this mirror node to fourth input of the **Randomise** operator. The node tree should look like this:



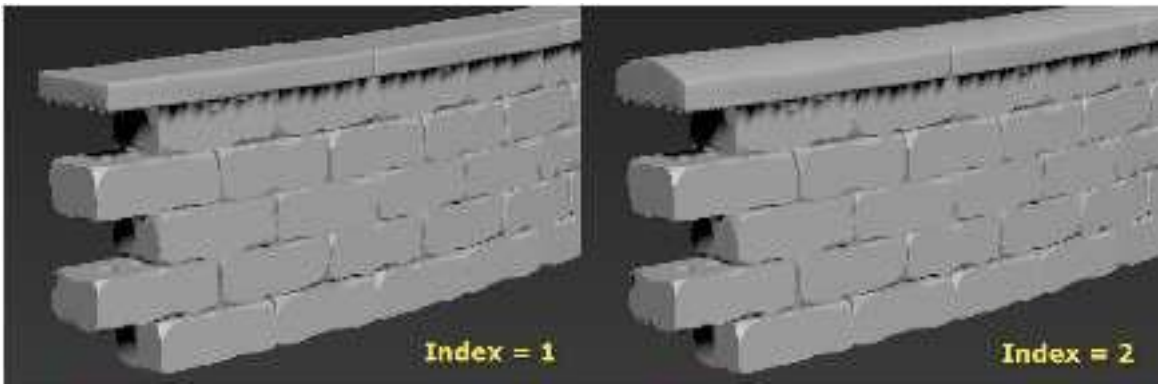
Exercise: Allow the user to select the cap type

In this section we'll use the Selector operator to create a style that gives the use the choice of two caps for the wall. Carry on from the previous exercise and follow these steps:

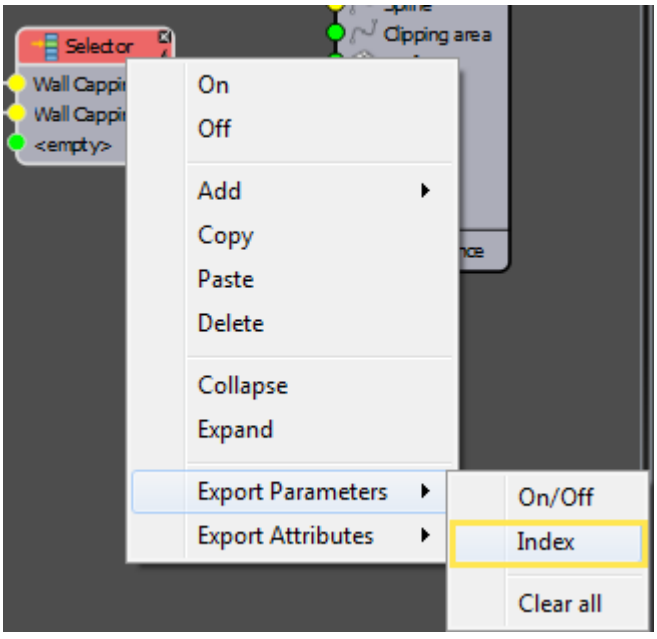
1. Drag a new **Selector** operator from the **Items list** to the **Construction View**.
2. Wire the segment called Wall ***Capping 01*** to the **Selector** operator's **first** input.
3. Wire the Segment called Wall ***Capping 02*** to the **Selector** operator's **second** input.
4. Wire the **Selector** operator to the **Cap** generator's **Default** input. The node tree should look like this:



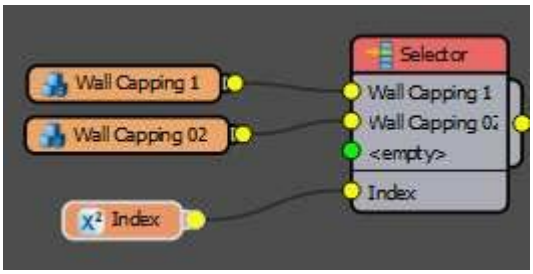
5. To switch between the cap types, click on the **Selector** node and go to the **Properties** panel. From here you can change the **Index** to select from the two cap types.



6. Though this works, it's not ideal to have to open the Style editor to change the Index value. To make it possible to change the value from the Modify panel, you can export the Index value and attach a Numeric node. To do this, Right click on the **Selector** node and choose **Export Parameters** > **Index**



7. Create a new **Numeric** node from the **Items List** and wire it to the newly exported **Index input** on the **Selector** node.



8. Finally, click on the Index node and go to the **Properties** Panel. Turn on limits and enter a **Minimum** value of **1** and a **Maximum** value of **2**. This is to ensure that you can only pick one of the 2 attached cap segments.

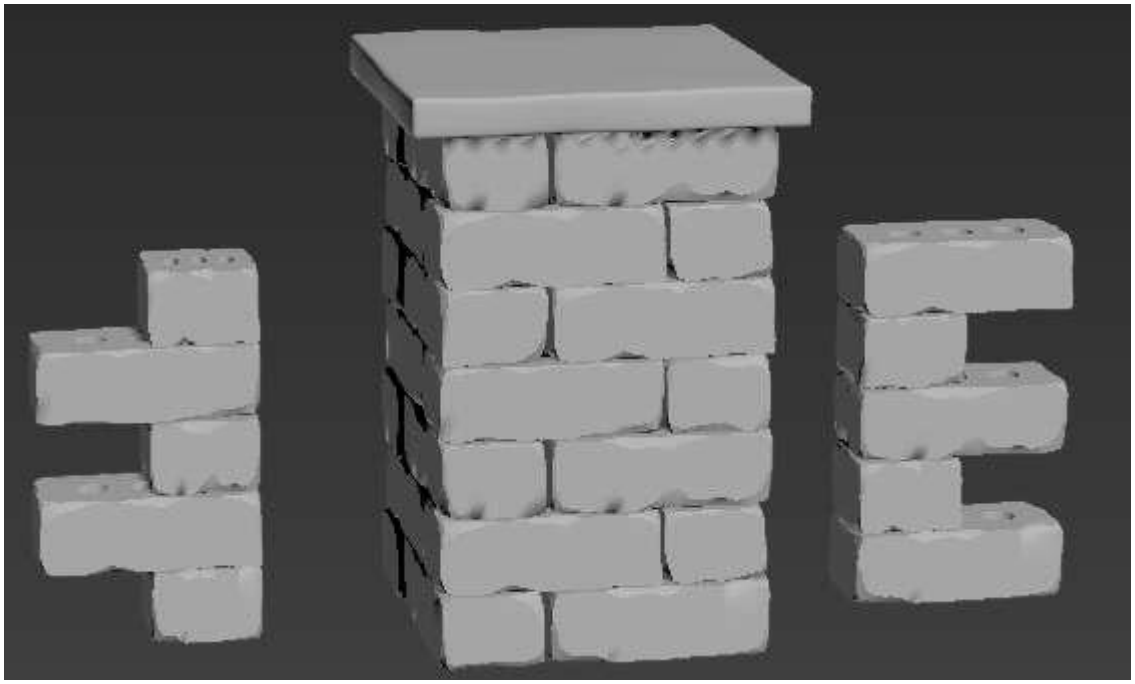
You now have a wall with randomised segments for the default section and a choice of two cap types. To finish the style we need to add the piers and the start and ends of the walls.

12.2 The Compose operator

The Compose operator enables you to build a new geometry object by attaching several segments together. When combined the segments are distributed side-by-side along the X Axis. At first sight, the Compose operator is functionally similar to the Sequence operator that we'll examine in the next section, but the Compose operator will enable you to wire multiple segments to an input that normally would only accept a single segment such as the Start, End Corner and Evenly inputs of the L1S array, or the Corners, Inner Corner, Sides and X Evenly inputs of the A2S array.

Exercise: Using the Compose operator

In our wall example the Compose operator will be used to combine a brick pier and the pieces required to terminate the walls. The image below illustrates these segments in the order they will need to appear for the Corner and Evenly inputs. For the start and end inputs only two of the 3 segments are required.



To add a brick pier at the Start:

1. Create a new **Compose** operator by dragging it from the **Items List** to the **Construction View**.
2. Wire the ***Wall Pier*** segment to the **Compose** Operator's **first** input.
3. Wire the ***Wall Start*** segment to the Compose Operator's **second** input. (Remember the segment values have already been added to ensure the segments dovetail together correctly)
4. Wire the Compose operator to the **Wall and Pillars** Generator's **Start** input.

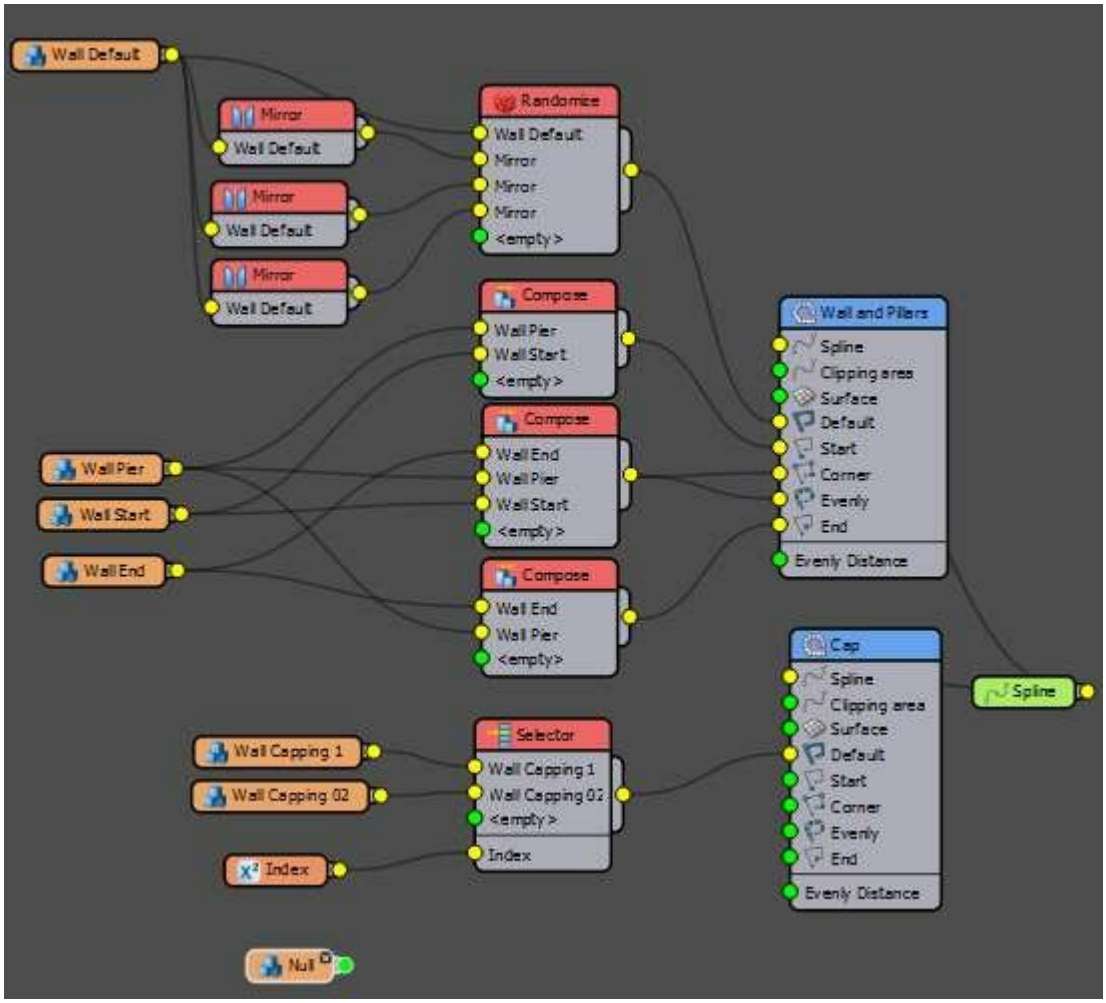
To add a brick pier at the End:

1. Create a second **Compose** operator by dragging it from the **Items List** to the **Construction View**.
2. Wire the ***Wall End*** segment to the **Compose** Operators **first** input.
3. Wire the **Wall Pier** segment to the **Compose** Operators **second** input. Note that this is the reverse order of the Compose used for the start input
4. Wire this Compose operator to the **Wall and Pillars** Generator's **End** input.

To add a pier at regular intervals and Corners:

1. Create a third **Compose** operator by dragging it from the **Items List** to the **Construction View**.
2. Wire the ***Wall End*** segment to the **Compose** Operators **first** input.

3. Wire the ***Wall Pier*** segment to the **Compose** Operators **second** input.
4. Wire the Wall Start segment to the Compose Operators **third** input.
5. Wire this Compose operator to the **Wall and Pillars** Generator's **Evenly** and **Corner** inputs. The finished node tree will look like this:

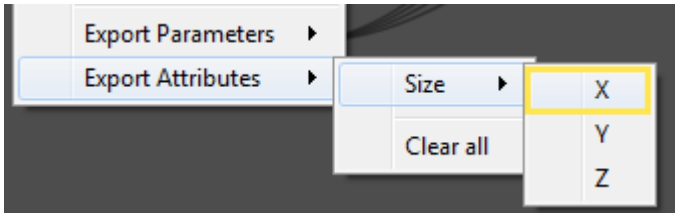


If you look at the caps now, you'll notice that they go through the brick pillars. This is because they run to the end of the spline, whereas the low brick wall is offset by the pillars. We can add an Empty segment with a fixed size to the Caps generator to compensate for the size of the pillars and push the capping sections away from the ends, corners and evenly segments in the style. In addition, in this section instead of entering the size of the pier manually, we'll export the X Size attribute of the Pier segment and use this value to automatically create the correct spacing. To do this:

> An empty or null segment contains no geometry of its own but can be very useful for adding spaces to the array.

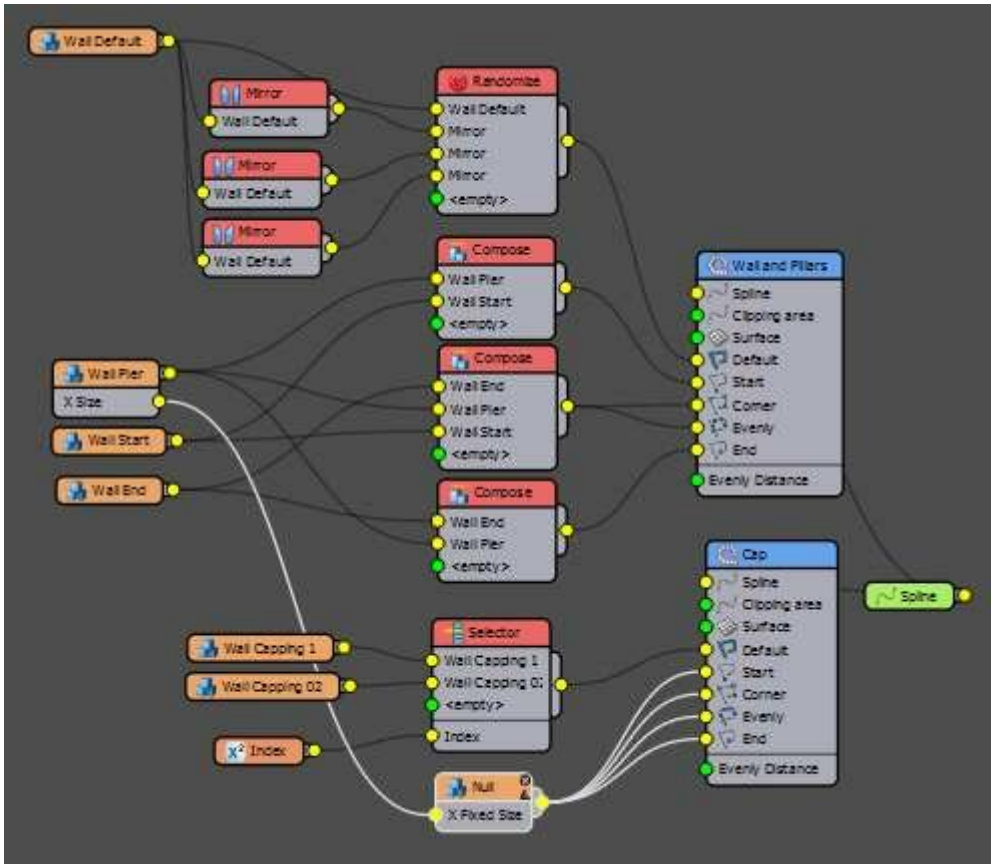
1. Wire the **Null** segment to the **Cap** generator's **Start**, **End**, **Corner**, and **Evenly** inputs.
2. Right click on the **Null** segment and select **Export Parameters** > **Fixed Size** > **Fixed Size** > **X**

3. Right Click on the ***Wall Pier*** segment and select **Export Attributes** > **Size** > **X**



4. Wire the **Wall Pier** Segment's **X Size output** to the **Null** segment's **X Fixed Size input**.

The final node tree looks like this ...



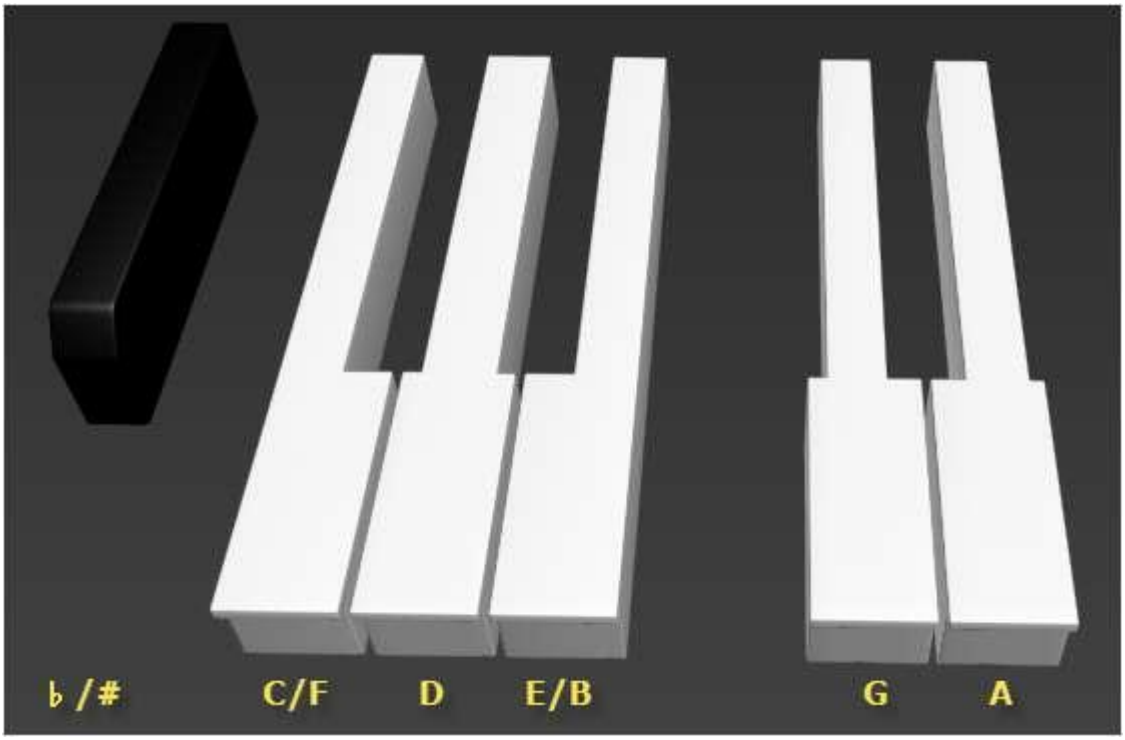
...which creates this style:

12.3 The Sequence operator

The Sequence operator creates patterns of segments by cycling through the list of inputs, starting at the top of the list and working downwards. Each segment can be repeated using a simple counter, when the counter reaches its limit, RailClone moves on to the next item in the list. When the sequence reaches the end of the list, it starts again from the beginning. Unlike the compose operator, a sequence can increment on either the X or Y axis and a sequence can be reset at either the start of each spline or the start of a new section.

> In RailClone a section is defined as a length of path found between two segments of the Start, End, Corner, or Evenly type

To illustrate this operator we'll recreate the pattern found on a piano Keyboard. This style uses the six segments illustrated below. To save time these have already been added to a RailClone style and padding values have been added to ensure that the keys fit together correctly.



Exercise: Using the Sequence operator to create a piano keyboard

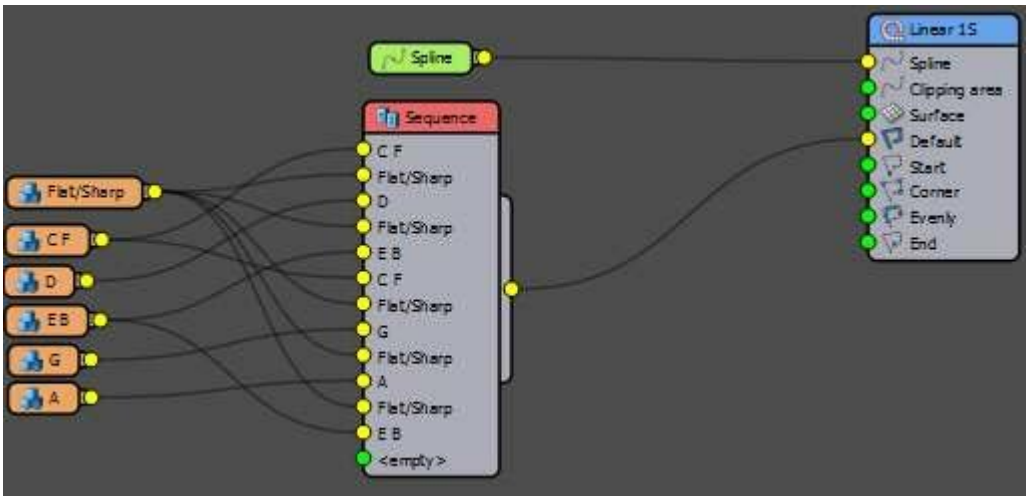
1. Open the file named sequence_piano_start.max and select the RailClone object from the scene called ***rc_piano***.
2. Go to the **Style** rollout and click

   to open the **Style Editor**. As you can see the segments for all the keys have already been added to the style.
3. To create the pattern, add a new **Sequence** operator by dragging it from the **Items list** to the **Construction View**.
4. Wire the **Sequence** operator to the Generator's **Default input**.
5. Wire the key **Segments** to the **Sequence** operator in the following order (from the top-down)
   - **C/F**
   - **Flat/Sharp**
   - **D**

- **Flat/Sharp**
- **E/B**
- **C/F**
- **Flat/Sharp**
- **G**
- **Flat/Sharp**
- **A**
- **Flat/Sharp**
- **E/B**

6. The final node tree will look like this:



That's it! - This sequence operator creates the repeating 12 note pattern found on a piano. The sequence operator is ideal for creating regular and repeating patterns on the X or Y axis.

12.4 Related Tutorials

Random Books Tutorial

This tutorial explains how to use the Randomise operators to select from 11 book segments.

Parquet Floor Tutorial

The chevron and herringbone patterns both use Compose operators to create their patterns. Follow these tutorials to see another way to use the Compose operator.

Stadium Tutorial Part 5

Part 5 of the Stadium tutorial uses Randomise and Mirror to create an audience with flags.

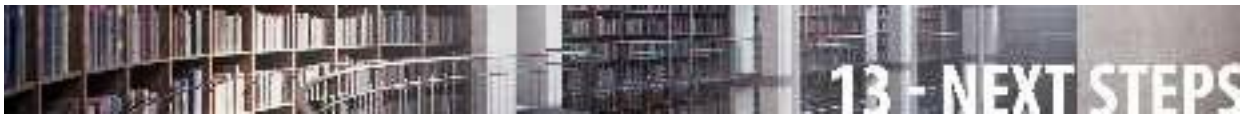
Create a seaside promenade

For a similar usage of the Compose operator as seen in the Brick wall example on this page, check out the balustrade example explained in the Seaside Promenade tutorial.

Documentation

For more information please see the Operators and Exporting Parameters sections of the online documentation.

13 Next steps



Congratulations! You've reached the end of **Getting Started with RailClone**. If you followed this guide and completed the exercises, you should now have a good understanding of the principles that underpin RailClone's approach to procedural modelling, You'll be able to to load and adapt presets from the library, create your own 1d and 2d arrays, and use operators to modify segments.

To continue to get the most out of RailClone, we have a number of additional resources to help you develop your skills further.

13.1 Tutorials

We release regular tutorials that range from explanations of basic concepts, to advanced topics and demos of possible applications for RailClone. We've suggested relevant tutorials throughout the guide but to see the full list, check out the tutorials page. Come back regularly, we're adding new content all the time!

13.2 Intermediate guide

We'll shortly be releasing the Intermediate Guide to RailClone. Once you've completed the Getting Started guide, this is intended as the next step. We'll build on the foundation you've learned here and introduce topics such as modelling tips for RailClone, wiring parameters, dealing with material ids, using surfaces, clipping splines, spacing modes, creating conditional relationships and much more. This will be followed by an Advanced Guide that's focused on writing Arithmetic Expressions.

13.3 Documentation

For a thorough explanation of all of RailClone's features, including procedures for many common operations, take a look at the Documentation.

13.4 Gallery

Get inspiration from some of the excellent work submitted to our gallery, and if you create an outstanding image or animation with RailClone (or Forest Pack) we'd love to see it. We're always on the look out for high quality material to feature on our gallery and social media , so if you have work you think meets the brief, please send your renders to submission@itoosoft.com , or upload them to our forum . We'll share them on our networks with full credit and links to your website.

We hope you enjoy using RailClone and wish you the very best with your work. If you have any feedback, queries or suggestions please let us know.