ConnectHandle and OrientHandle

This document explains the principles and usage of the two Rhino Commands in the plugin.

Orientation principles

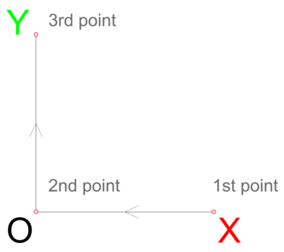
Placing an object in space requires not only its location (as XYZ coordinates) but also information about its orientation; all of these can be expressed by an *Orientation Plane*. Imagine having an object attached to the palm of your hand, with your hand in an horizontal position, palm up: your hand is the XY Plane, with the Z axis pointing out of the palm. To move the object around you have to move your hand in a different position and orientation. This operation coincides with the definition of a destination Plane. A transformation that moves an object from one plane to another is called an Orient Transformation.

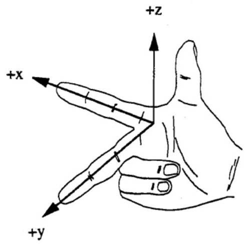
Connection principles

In case we want to assemble 2 objects, we cannot simply overlap their orientation planes with Orient: it would be the equivalent of holding both objects in the palm of your hand, sharing the same space (which is digitally allowed, but violates certain laws of physics....). In this case the planes need to face or "kiss" each other: they share the same origin point but their Z axis are in opposing directions.

Geometric Principles

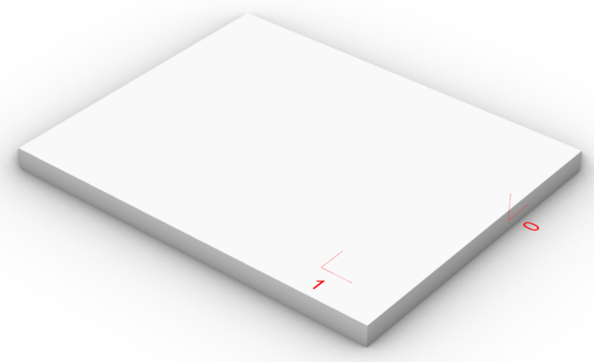
ConnectHandle and **OrientHandle** are commands that perform Plane-to-Plane Transformations using "handles" (L - shaped polylines). A "handle" identifies a Plane by defining its Origin point as well as its X and Y axis. The Z axis is determined by right-hand rule.





Handles polylines ***must*** be drawn like in the above figure (respecting points order and the 90° angle). In this case, the Plane's Z axis is pointing out of the screen, towards you.

For an assemblage, Handles can be placed on an object to identify connector Plane locations, with the Z axis always pointing outwards. The object in the following image (from the example file included in the plugin) has 2 handles, 0 and 1:



Command Steps

1. Type **_ConnectHandle** or **_OrientHandle** in the Rhino Command Window
2. Select the object(s) you want to connect/orient
3. Select starting (sender) handle
4. Select destination (receiver) handle
5. Input a rotation angle (in degrees)
6. The command displays a preview - you can accept or change the rotation angle

The commands create copies of the oriented objects.

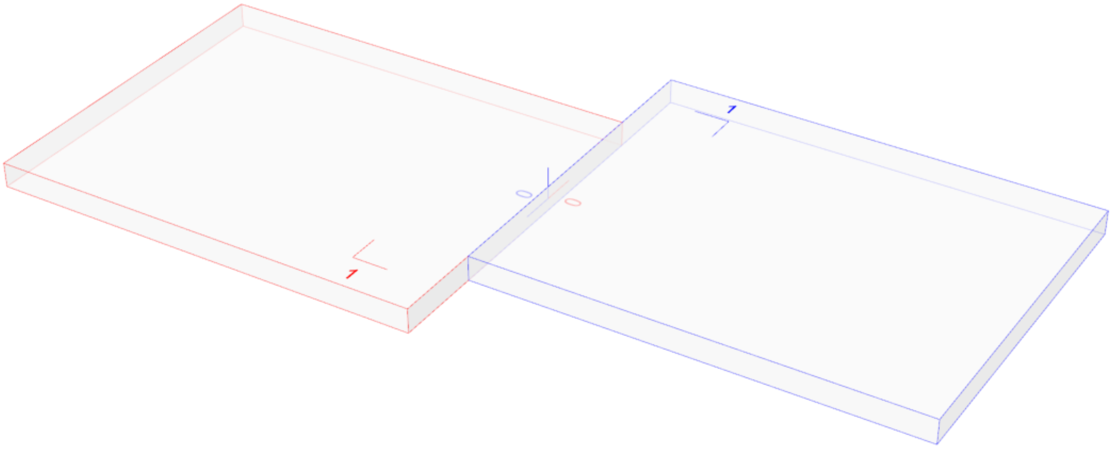
Basic Examples

ConnectHandle

Let's say we want a copy of the object above and connect both of them at their connector 0, no rotation. First of all, group together the object and its handles (select all the objects you want and type _Group), so that also the handles will be oriented. Then:

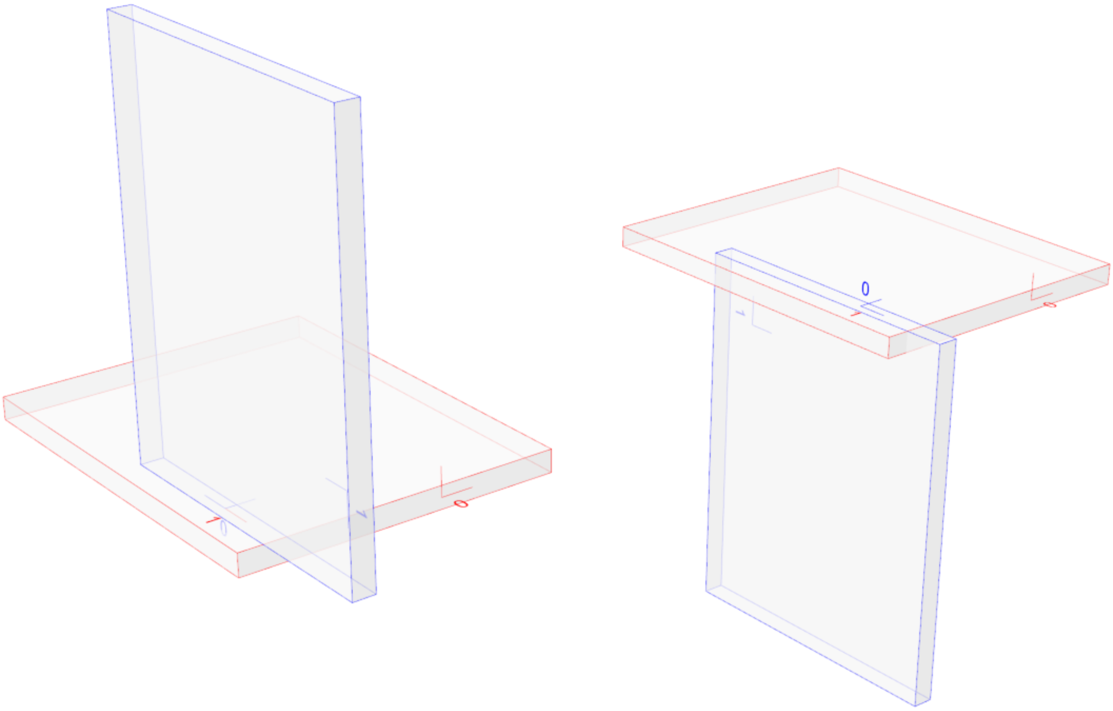
1. Type ***_ConnectHandle*** in the Command Window (you can omit the _ if your Rhino is in English)
2. Select the newly created group and hit enter
3. Select the polyline at index 0
4. Select again the same polyline
5. Input 0 as rotation angle
6. Accept the preview (type Yes, Y or click the Yes option)

This is what you get:



Connect vs Orient

Let's see the difference between **ConnectHandle** and **OrientHandle**. This time, we'll use handle 0 as sender and handle 1 as receiver:

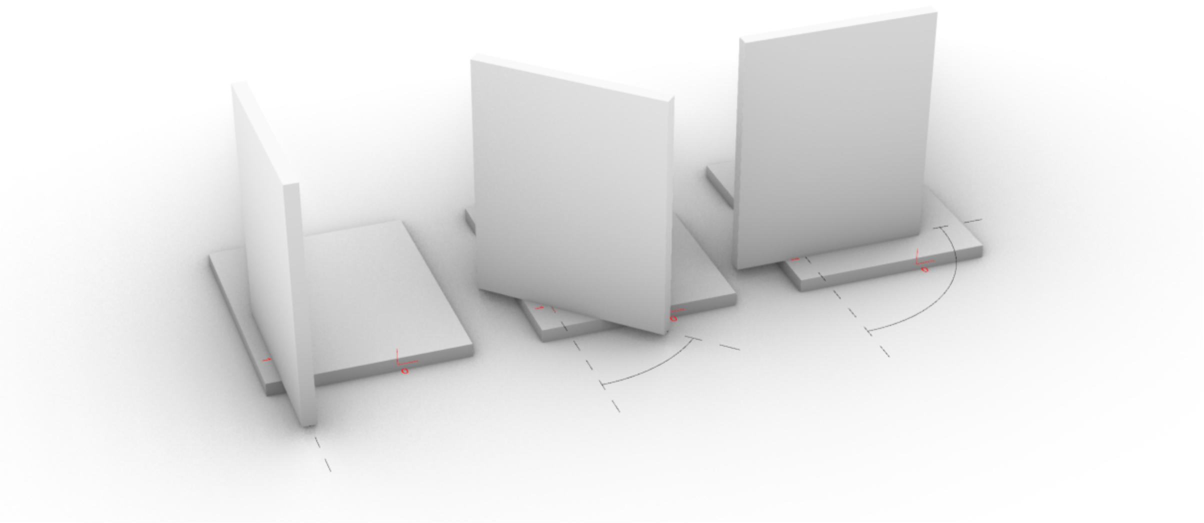


ConnectHandle (left) vs OrientHandle (right)

OrientHandle works as a canonical Orient Transformation, aligning the sender plane with the receiver (had we chosen to orient handle 0 on itself, the object would just have ended up on the same initial position). **ConnectHandle** makes the two planes "kiss" each other, aligning the objects properly. In the case of assembling objects together, **ConnectHandle** is our weapon of choice.

Rotation angle

Now try **ConnectHandle** from handle 0 to 1, but with rotation angles different from 0 and see what happens. Rotation angles are measured in degrees, progressing counterclockwise around the Z axis of the receiving handle.



ConnectHandle from handle 0 to handle 1, rotation angles of 0, 45 and 90 degrees

Hints

- ***Using Blocks*** - For objects with complex structure (many layers, different geometries, details, etc.), the best option is to create **Blocks**. In this case, ***DO NOT INCLUDE HANDLE POLYLINES in the block***, or they won't be selectable. Instead, create a block with all the geometry except the handles polylines, then Group the block and the handles together. You can label handles with text or text dots for convenience.
- ***Purge empty Groups*** *(from time to time)* - Rhino collects information on each newly created Group; groups are identified by name (unless you manually rename then with *_SetGroupName*) and you can see how many groups have been created with the *_SelGroup* command. Group information persists even if the instances of the Group are deleted. To avoid unnecessary large files, run the Purge instructions from time to time and make sure to check the Groups option.