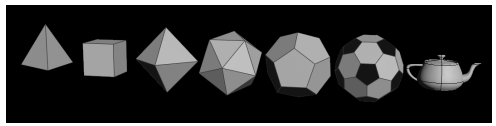
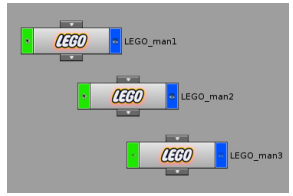
INTRODUCTION

A **Houdini Digital Asset** (**HDA**) is the culmination of a procedure that **converts an entire node network** into a **single operator**. This HDA is then **saved externally** from the Houdini environment and can be **utilised in any other scene** as either **an object** in its own right, or as a **custom operator** in another network. Digital Assets can be **created at any level** of Houdini and can be easily **shared between different users**. They can also be **modified** and **refined** after their creation, and can even **contain** other **Digital Assets**.



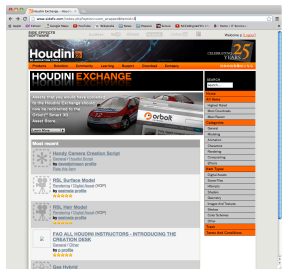
The Platonic Solid SOP is one of **many pre-made Digital Assets within** the **Houdini** environment. It can switch between various pre-defined shapes (tetrahedrons, Icosahedrons, Utah Teapots etc). Its **status** as a **HDA** is revealed by its **blue node name**. This **indicates** that there is an **internal network** that can be **examined** by going **inside the node**. By default, this **internal network** is **locked** to prevent its modification, but **all Digital Assets** can be **unlocked** when internal editing or modification is required.

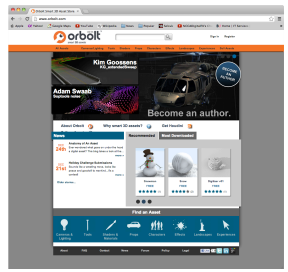
Digital Assets are a great way for **sharing common resources** in **group-based working**. Digital Assets are also **valuable learning tools**, as **internal networks** can be **examined**, **deciphered** and **replicated**.



Digital Assets can also **speed up production work** where **multiples** of a specific event are required. For example, an **army of differing LEGO Men** can be **generated** from a **single LEGO Man Digital Asset** where **elements** such as **clothing**, **hats**, and **faces** can all be **procedurally controlled** on a **per instance** basis.

Digital Assets can also be **uploaded** and **shared** with other **Houdini Artists** via the **Houdini Exchange** (www.sidefx.com/exchange) or the **Orbolt website** (www.orbolt.com). These are online libraries of Digital Assets kindly generated, shared and uploaded by the wider Houdini community working in Production.

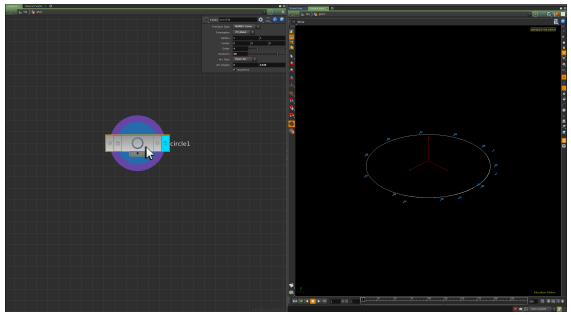




Digital Assets can also now be **ported** into **other software** such as **Maya**, **Unity**, **Unreal** and **Cinema 4D** thanks to **Houdini Engine**:
http://www.sidefx.com/index.php?option=com_content&task=blogcategory&id=227&Itemid=381

CREATING A SPIRAL GENERATOR DIGITAL ASSET

This exercise will look at the creation of a custom operator specifically for generating spirals, and will demonstrate the principles of HDA generation, modification, and design. In a **new Houdini scene**, create a **Circle Object**.



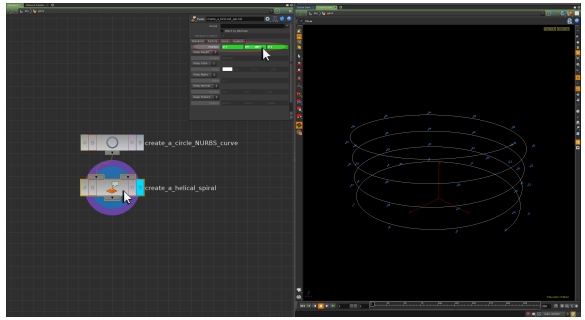
In the **parameters** of the **Circle SOP** specify:

Primitive Type	**NURBS Curve**	
Orientation	**ZX plane**	
Divisions	**40**	
Arc Type	**Open Arc**	
Arc Angles	**0**	**1440**

This will create a NURBS curve that loops upon itself four times.

Append to the **Circle SOP** a **Point SOP**. This operator will affect the curve on a per point basis. In the **Parameters** for the **Point SOP** specify:

Position	**$TX**	**$PT / $NPT**	**$TZ**



This will iterate through each point on the curve in turn and set its Y position to the current point number ($PT) divided by the total number of points ($NPT). Visually this will create a helical spiral that loops 4 times (due to the 1440 end arc angle specified in the Circle SOP).

NOTE: The **Point SOP** variables **$TX**, **$TY**, and **$TZ** are essentially 'empty' variables that collect and store the point positions of any input geometry. The variables **$PT** and **$NPT** are local to the **Point SOP** and documented in its **Operator Help Card**.

CREATING A SECOND SPIRAL

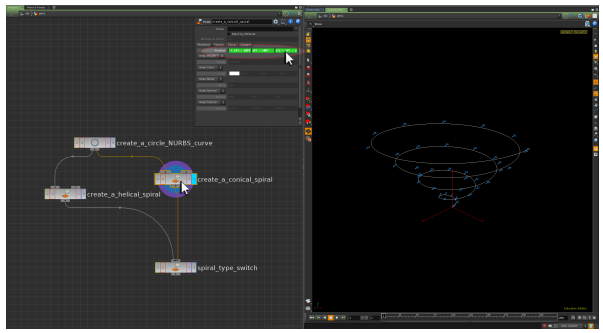
Copy (**Ctrl + c**) and Paste (**Ctrl + v**) the **Point SOP** to create a second one. Modify the Position parameter from:

Position	**$TX**	**$PT/$NPT**	**$TZ**

to

Position	**$TX * $PT/$NPT**	**$PT/$NPT**	**$TZ * $PT/$NPT**

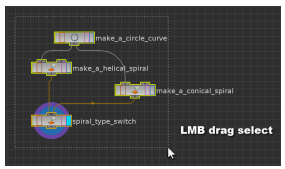
This will change the helical spiral into a conical spiral. Rename the two Point SOPs to **helical_spiral** and **conical_spiral** respectively and wire them both into a Switch SOP.

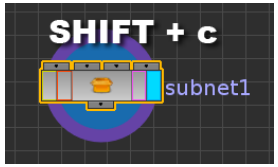


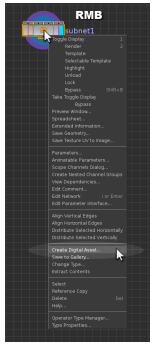
The Switch SOP will be utilised to switch between the two spiral types.

CREATING THE DIGITAL ASSET

It is possible to generate a Digital Asset even with this simple network. The advantage of creating the Digital Asset now, is that it will allow the Asset to be tested as well as allowing a more complex network to evolve inside it. In the **Network Editor**, **select all the nodes** and press **SHIFT + c** on the keyboard. This will collapse all of the nodes into a sub network from which the Digital Asset can be generated.



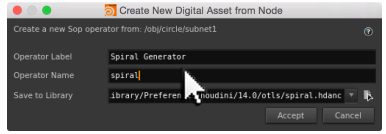




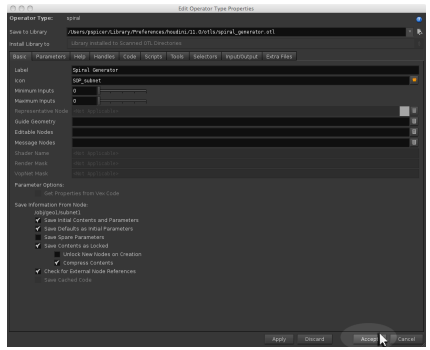
RMB on the subnet, and from the resulting menu choose **Create Digital Asset…**

In the resulting window specify:

Operator Label	**Spiral Generator**
Operator Name	**spiral**
Save to Library	**$HOME/houdini14.0/otls/spiral.hdanc**



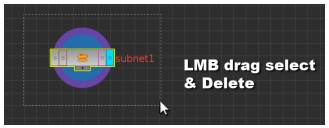
The **Operator Label** is the longer operator **name** Houdini will list in the **TAB menu system**. The **Operator Name** is the **name** of the custom node. The **Save to Library parameter** gives the **destination** of the **Houdini Digital Asset** (.hda or .hdanc – non commercial) where it will be saved. It should automatically re-name itself relative to the Operator Name declaration. With these options specified, press **Accept**.

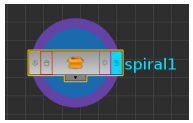


Doing this will also activate the **Edit Operator Type Properties** window. This window will be **used** to **generate** the **end user interface** of the Digital Asset. For the moment however press **Accept** button to **close this window**. See **spiral_generator_stage1.hipnc**

ACTIVATING THE DIGITAL ASSET

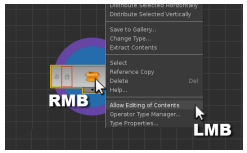
As the Digital Asset has now been created, it can be activated as a new operator. In the Network View, delete the subnet1 node. In its place press **TAB** and **type Spiral Generator**.

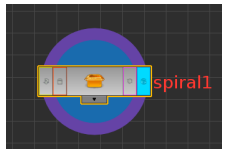




UNLOCKING THE DIGITAL ASSET

When a **HDA node is activated**, it is **locked** by default. This means it is not possible to edit or modify the internal network. The blue operator name label visually indicates this. A HDA can be **unlocked** by **RMB** on the node icon and from the resulting menu choose **Allow Editing of Contents**. The operator name label will turn red indicating its editable status.





This process of activating and unlocking the Digital Asset can be done at any stage of its development to help ensure the asset is working as expected.

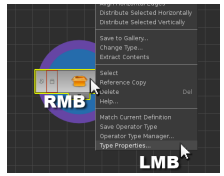
NOTE: For this process to work correctly, the **Edit Operator Type Properties window** must be **closed**.

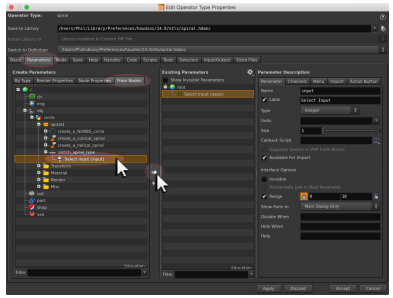
At the moment there are **no End User controls** for the Digital Asset, and its **Parameter window** is **empty**. **End User parameters** for Digital Assets **can be created** and activated **using the Edit Operator Type Properties window**. Digital Assets must be unlocked to allow for parameter creation and activation.

CREATING THE SWITCH SPIRAL TYPE PARAMETER

When configuring Digital Assets, care should be taken over the design of the End User Interface to ensure only the core controls for the asset are activated in a clear, coherent and logical way. The first End User control that can be activated is a **parameter** for **switching between** the **two spiral types**. As this mechanism is already **established** as part of the **internal network**, it can be **ported up directly** from the **spiral type Switch SOP**.

RMB on the unlocked Spiral Generator node, and from the resulting menu choose **Type Properties…** This will re-activate the **Edit Operator Type Properties** window for this Digital Asset.





Using the **From Nodes** section of the **Create Parameters** list, **unstow** the **Select Input** parameter from the **switch_spiral_type** node. With this parameter selected, port it across to the **Existing Parameters** list using the **right facing arrow**. With this done, press the **Apply** button to see the effect of this action.

THE DIFFERENCE BETWEEN ACCEPT AND APPLY

If **Apply** is pressed in the **Edit Operator Type Properties** window, the **changes** are **instigated** with the **window remaining open**. If **Accept** is pressed, the **changes** are **instigated** and the **Edit Operator Type Properties window closes**. If the Edit Operator Type Properties dialog box inadvertently gets closed, it can be **reactivated** by **RMB** on the **Digital Asset node** and choosing **Type Properties…** from the resulting menu.

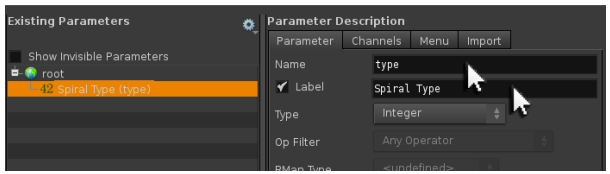
MODIFYING THE PARAMETER FUNCTIONALITY

When the Parameters pane for the Spiral Generator is examined, the **Select Input slider** can be seen. When the **slider** is increased from **0** to **1**, the switch mechanism inside the Digital Asset is triggered and the different spiral types can be seen in the Viewer.

While the switching mechanism itself is working, the **End User control** for it could be **more intuitive**. As this **parameter** is **integer based**, the **Edit Operator Type Properties** window can be used to **convert this parameter** from a slider **to a drop down menu**.

CREATING A MENU

Reactivate the Edit Operator Type Properties window, and under the Parameter Description section set the internal parameter **Name** to **type** and the external parameter **Label** to **Spiral Type**. This will create more intuitive names for this Digital Asset Parameter.



In the **Parameters Description > Menu** section, activate the **Use Menu** toggle, and in the two text entry fields underneath the Menu Items listing, enter the following respectively and press **ENTER**:

0 **Helical**

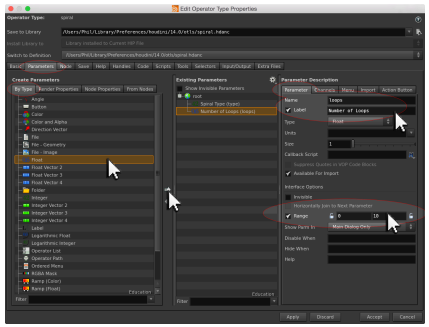
1 **Conical**

This will create a drop down menu system for the spiral type where the integer values of 0 and 1 are replaced by the words Helical and Conical. With this set, the **Accept** button can now be pressed. When the Parameters Pane for the Spiral Generator is re-examined, the integer slider has been replaced by a drop down menu describing each spiral type.

CREATING A CUSTOM PARAMETER

Another **useful End User control** is the ability to **control** the **number of spiral loops**. At present the **internal network** for the spiral **generates 4 loops**, although there is no explicit parameter controlling this. Instead a **custom parameter** can be created at the **top level** of the **Digital Asset**, and then **fed back** into the **internal network** for controlling the number of loops being generated.

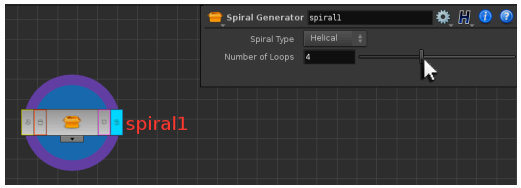
Reactivate the **Edit Operator Type Properties window**, and from the **Create Parameters By Type** list, select a **Float parameter** and using the **right facing arrow**, **port** it over to the **Existing Parameters list**.



In the **Parameters Description > Parameter** section specify:

Name	**loops**
Label	**Number of Loops**
☑	**Range**

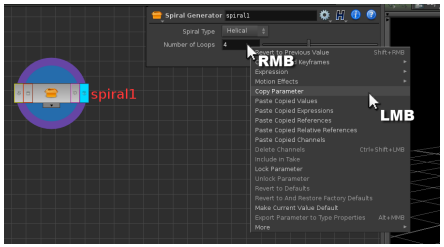
Under the **Parameters Description > Channels** section of the window, specify a **Defaults** value of **4** for this new parameter.



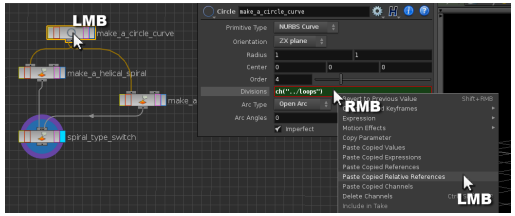
With this in place, the **Accept** button can be pressed. Now the spiral generator Parameters has a **currently unlinked slider** that can be utilised to control the number of spiral loops.

ACTIVATING THE CONTROL

RMB on the newly added parameter and choose **Copy Parameter**.



Go inside the Spiral Generator internal network, and select the Circle SOP.



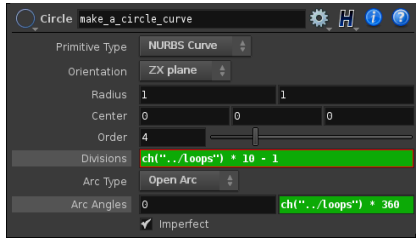
RMB over the **Divisions** Parameter and choose **Paste Copied Relative References**. This will create a path expression linking the Number of Loops parameter value to the number of points creating the NURBS circle. Modify this path expression from

ch("../loops") to **ch("../loops") * 10 – 1**

This will **ensure** that if the **Number of Loops Parameter** has a **value** of **3**, **30 points** will be **generated**. If the **value** is **4**, then **40 points** will be **generated**, and so on. This can be **verified** by **MMB** on the **Circle SOP** node.

Locate the **End Arc Angles parameter** of the Circle SOP (currently specified as 1440), and **RMB** over it to again choose **Paste Copied Relative References**. **Modify** the **Channel Reference** expression to read:

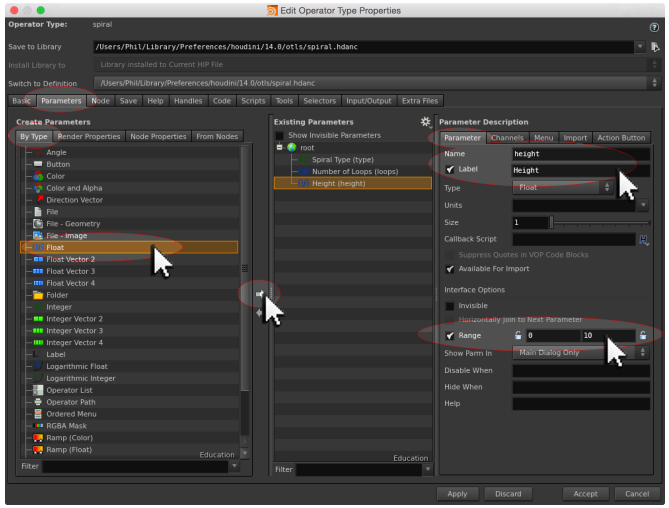
ch("../loops") * 360



As the Number of Loops parameter is currently returning a value of 4, multiplying it by 360 will return an End Arc Angle parameter value of 1440. The number of spiral loops and also the number of points constructing the spiral, are now controlled directly by the Number of Loops parameter located at the top level of the Digital Asset.

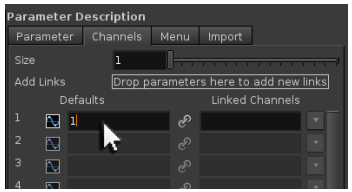
CREATING A HEIGHT CONTROL

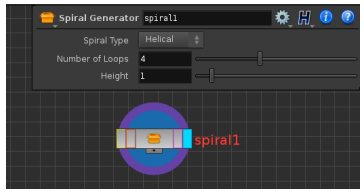
A custom parameter can also be used to control the height of the spiral. A default height of 1 is given to the spiral through the expression $PT/$NPT found in the Point SOPs creating the different spiral types. This value can be modified to include a height multiplier parameter set at the top level of the Digital Asset.



Reactivate the **Edit Operator Type Properties** window, and as before **port** across a new **Float** parameter. Under the **Parameter Description** section specify the **Name** of this new parameter as **height** and the **Label** as **Height**. Activate a **Range** for this parameter.

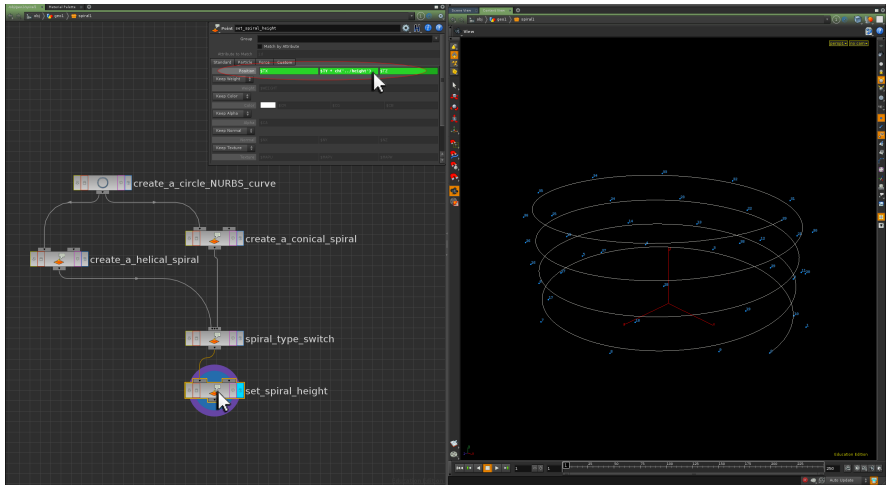
Under the **Channels** section of the **Parameter Description** area, set a **Default Value** for the **Height** parameter of **1**. When the **Accept** button is pressed, the Height parameter slider can now be seen on the Spiral Generator Digital Asset.





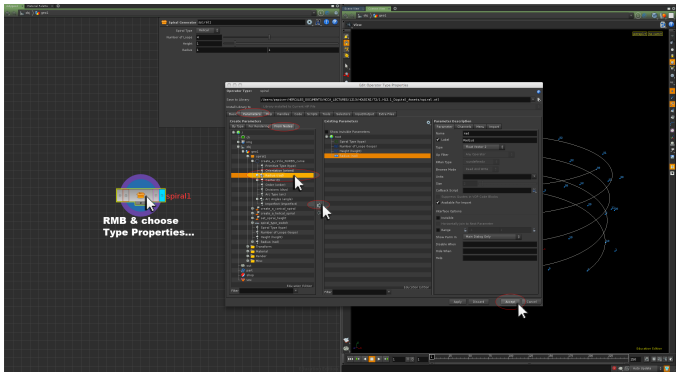
The **Height parameter** can be **activated** by returning back to the **internal network**, and appending a **new Point SOP** to the **end** of the **spiral network**. In the **parameters** for this new Point SOP specify

Position	**$TX**	**$TY * ch("../height")**	**$TZ**



CREATING A RADIUS PARAMETER

RMB on the **Spiral Generator node**, and **reactivate** the **Type Properties** window. **Under** the **Create Parameters from Nodes** List, locate the **Radius parameter** of the Circle SOP node, and **port it across** to the **Existing Parameters List**. When **Accept** is pressed, the Radius parameter will be active at the top level of the Digital Asset.

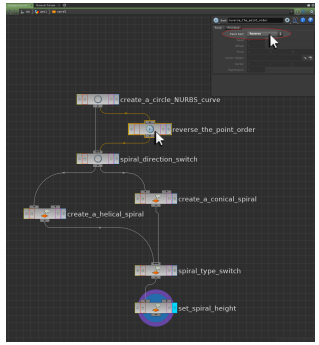


The Height parameter will now increase or decrease the height of the spiral in the Viewer. Similarly the Radius parameter can be used to adjust the X and Z size of the spiral. See file **spiral_generator_stage2.hipnc**.

REVERSING THE SPIRAL

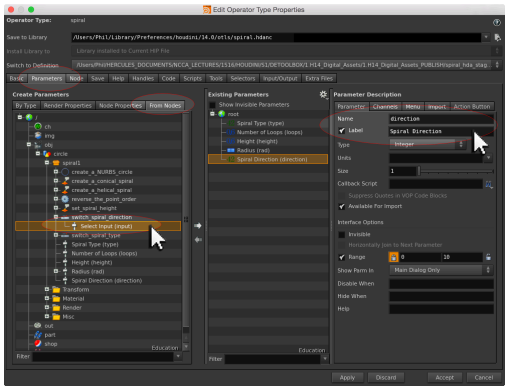
Another **useful parameter** for the Spiral Generator might be to **give** the **end user choice** over the **spiral direction**. Go back into the internal network for the spiral generator, and **RMB** append a **Switch SOP** to the **Circle SOP**.

Rename this Switch SOP to **spiral_direction_switch**. **MMB** on the output of the **Circle SOP** to create a **Sort SOP** as a separate chain. Pipe the output of the **Sort SOP** as the second input into the **Switch SOP**. Rename this Sort SOP to **reverse_the_point_order**, and set the **Point Sort parameter** of the **Sort SOP** to **Reverse**.



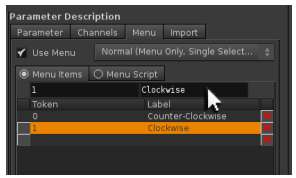
NOTE: Whenever a **network** is **modified**, the **Display and Render Flags** must be **set back** to the **final node in the network**, so it will become the default whenever the Digital Asset is resaved.

Reactivate the **Edit Operator Type Properties** window for the Spiral Generator, and from the **Create Parameters > From Nodes** section, identify the **Select Input** parameter found inside the **spiral_direction_switch**. Port this parameter across to the **Existing Parameters** list, and give it a new **Name** of **direction**, and a new **Label** of **Spiral Direction**.



Under the **Parameter Description > Menu** section, activate the **Use Menu** toggle option, and specify in the **Menu Items**:

0	**Counter-Clockwise**
1	**Clockwise**

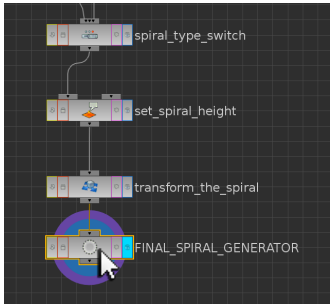


When **Accept** is pressed, a **drop down menu** for controlling the **spiral direction** can now be seen at the top level of the Digital Asset.

CREATING ADDITIONAL TRANSFORM CONTROLS

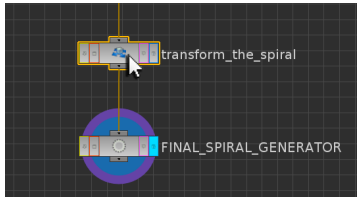
While the Spiral Generator is functioning, it cannot be positioned or resized. At present an End User would have to append a Transform SOP in order to move the spiral around and change its shape. While this is not overtly problematic as a workflow, it might be better to embed transform controls directly into the Spiral Generator itself.

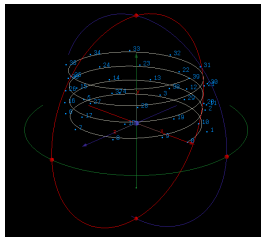
Activate the **Edit Operator Type Properties** window for the Spiral Generator. Inside the Network, append a **Transform SOP** and a **Null SOP** to the **set_spiral_height Point SOP**, and rename them appropriately.



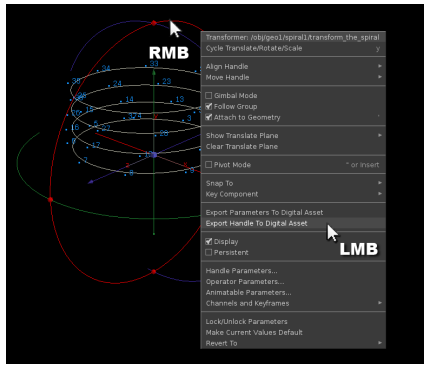
When the Edit Operator Type Properties window is **open** and **active** and the **asset** is **unlocked**; **additional parameters** can be **ported directly up** to the top level of the Digital Asset either through the **Parameter Pane RMB Menu** or through the **Viewer Pane Tool RMB Menu**.

Select the **Transform SOP** in the **Network Editor**, and press **ENTER** with the mouse over the **Viewer**.

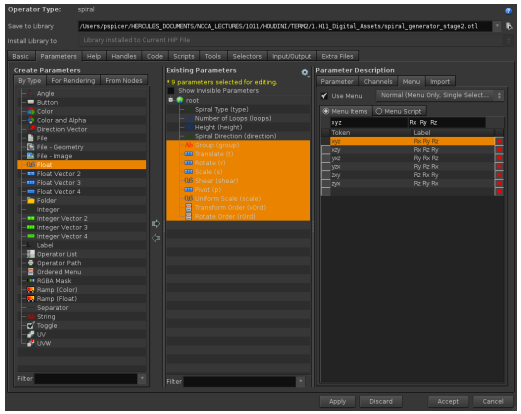




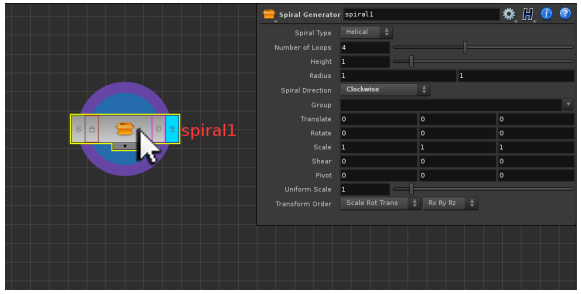
This will **activate** the **Tool Mode** for this operator and **reveal the interactive transform handles in the Viewer**. **RMB** on the **handle**, and from the **resulting menu choose Export Handle to Digital Asset**. This will port both the handle and its associated parameters directly to the Existing Parameters list as a single step.



When the **Existing Parameters list** of the **Edit Operator Type Properties** window is examined, the **transform controls** are now listed. Automatically created **handle listings** can also be examined under the Handles section of this dialog.



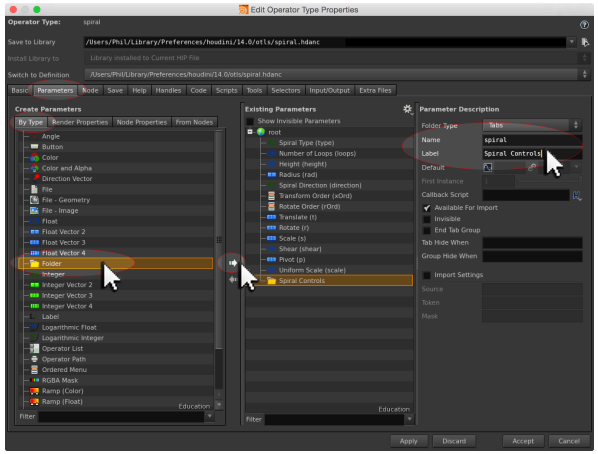
Press the **Apply** button, and then **minimise** the **Edit Operator Type Properties** window. When the **Parameters** of the **Spiral Generator** are examined, the additional transform parameters can now be seen. Also, the interactive transform handles can be activated and deactivated in the Viewer as part of the standard tool mode for the spiral generator.



TIDYING UP THE INTERFACE

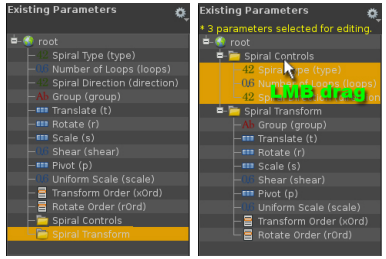
Some of these newly created parameters are useful to an End User; some are not (for example the Group parameter). **Extraneous parameters** can be **deleted** from the **Existing Parameters list** by selecting the **parameter** and pressing the **DELETE** key. **Relevant Parameters** can also be **LMB dragged up or down** in the list to reorder them appropriately to make the interface more intuitive.

In the **Edit Operator Type Properties** window, go to the **Parameters By Type** list and select **Folder**. Using the right facing arrow port the Folder over to the Existing Parameters list.

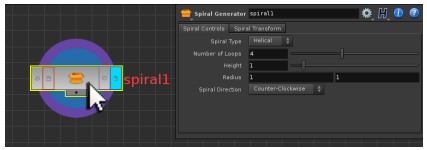


In the **Parameter Description** for the **Folder**, set the **Name** of the folder to **spiral** and the **Label** to **Spiral Controls**. Repeat this operation to create a folder called **Spiral Transform**.

LMB drag all of the **Spiral parameters** into the **Spiral Controls folder**. Repeat this, **LMB dragging** all of the **Transform parameters** into the **Spiral Transform folder**.

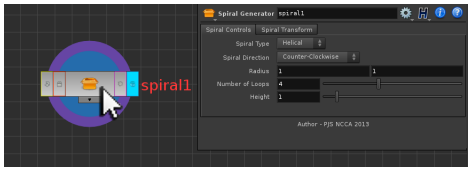


When **Apply** is pressed on the Edit Operator Type Properties window, the Spiral Generator parameters pane will update accordingly, with each set of parameters being stored within its own section.



ADDING LABELS

A label can be added to articulate the Digital Asset interface in a similar way to the creation of a folder. Labels can for example record both additional parameter information and HDA authorship.

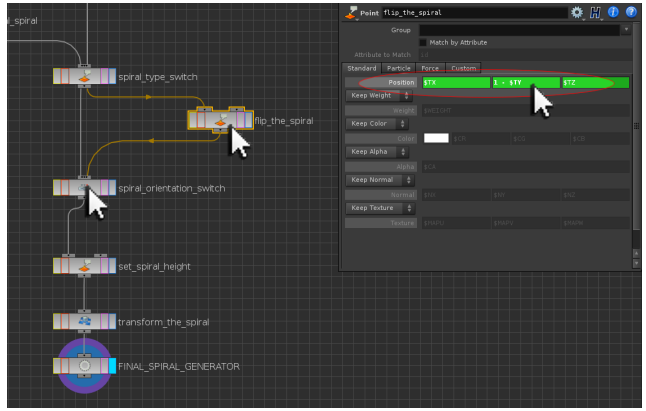


See file spiral_generator_stage3.hipnc.

ADDING FURTHER FUNCTIONALITY

The contents and functionality of an unlocked Digital Asset can be added to and refined after creation. This need for additional functionality may simply be to increase the flexibility of the tool or may be as a result of an End User request. In this example, it would be useful to flip or re-orientate the Conical Spiral, as well as allowing the user to convert the Spiral from a NURBS curve to a Polygon.

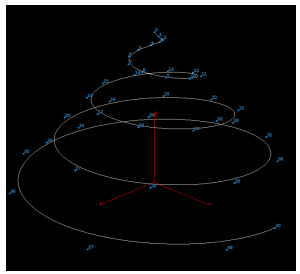
Using the Spiral Generator parameters, change the **Spiral Type** to **Conical**, and **activate** and **minimise** the **Edit Operator Type Properties** window. Inside the Spiral Generator network, append a new **Switch SOP** to the output of the **spiral_type_switch**. Rename this Switch SOP to **spiral_orientation_switch**.



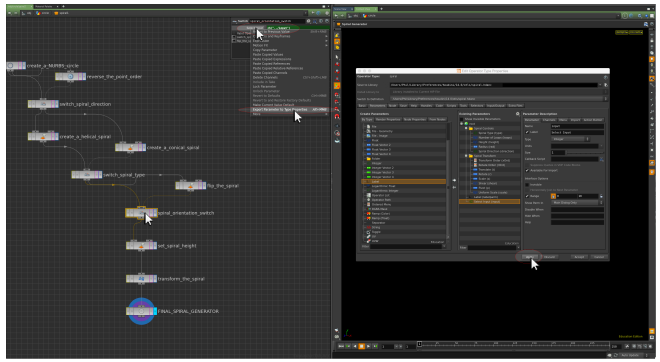
MMB on the output of the **spiral_type_switch** to insert a **Point SOP** as a new network branch. Rename this **Point SOP** to **flip_the_spiral**, and wire its output as the second input of the spiral_orientation_switch. In the **parameters** for the flip_the_spiral **Point SOP**, specify:

Position	**$TX**	**1 - $TY**	**$TZ**

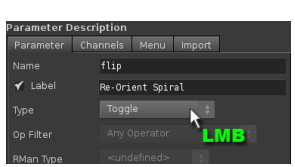
When the spiral_orientation_switch is activated, this will invert the conical spiral.

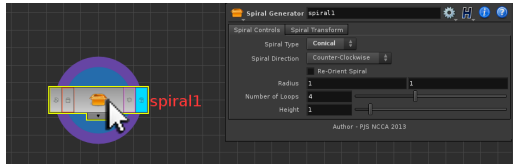


A higher-level parameter can be quickly created for this reorient function, by utilising the **Export Parameter to Type Properties** option. This can be found by **RMB** on the Spiral Orientation Switch's **Select Input** parameter **if** the **Operator Type Properties window** is **active**.



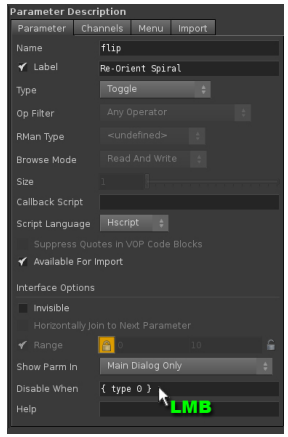
This control can be set as a toggle based option by changing the **Type** setting from **Integer** to **Toggle** in the **Parameter Description**. The **Name** and the **Label** of this parameter should also be modified accordingly.





DISABLING PARAMETERS

As the **Re-Orient toggle** performs a **similar operation** to the **Spiral Direction parameter** (when the **Spiral Type** is set to **Helical**), it **can** be **disabled** until the **Spiral Type** is set to **Conical**. To do this, the **Re-Orient's Disable When parameter** must be activated in the **Edit Operator Type Properties**:



Disable When **{ type 0 }**

This translates as saying; when the type parameter (Spiral Type) is set to its first input (0 – Helical), disable this flip parameter. When the Spiral Type parameter is now set to Helical, the Re-Orient parameter will no longer be active.

BOOLEAN DISABLING

This **Disable When** syntax can also form the basis of an **OR** list command which other parameters could be added to. This would disable the parameter when any of these listed parameters return the appropriate values.

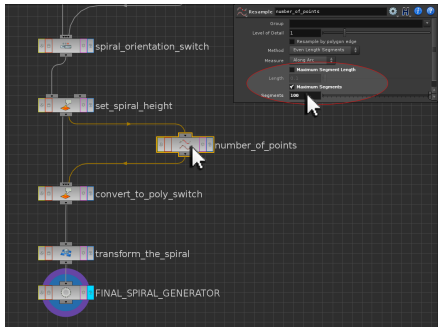
Disable When **{ parameterX 0 } { parameterY 5} {parameterZ 4}**

An **AND** list can also be created by specifying more than one parameter in a pair of braces. This would require both parameters to be set to the appropriate values before the disable function would work.

Disable When **{ parameterX 0 parameterY 5 }**

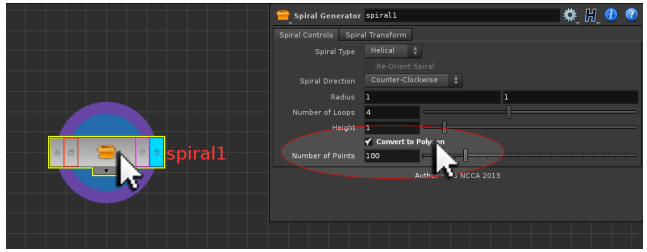
CREATING A POLYGON SPIRAL

RMB on the output of the **set_spiral_height Point SOP** to insert a **new Switch SOP**. Rename this switch from switch1 to **convert_to_poly_switch**. **MMB** on the output of the **spiral_orientation_switch** to insert a **Resample SOP** as a new network branch. Rename this Resample SOP to **number_of_points**, and wire its output as the second input of the convert_to_polygon_switch.



In the **parameters** for the **Resample SOP**, **deactivate** the **Maximum Segment Length** parameter, and **activate** the **Maximum Segments** parameter. A default setting of **100** will provide sufficient points to maintain the spiral shape when converted to polygon.

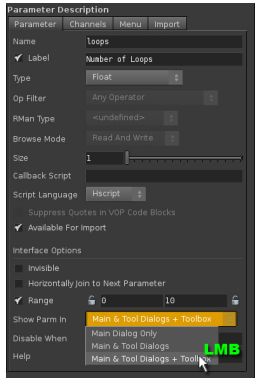
The **Convert to Polygon Switch** can also be **RMB** activated as a **toggle based** Digital Asset control. Disabling the Number of Points parameter when the convert to polygon toggle is not activated can be controlled as per the Re-Orient parameter example.

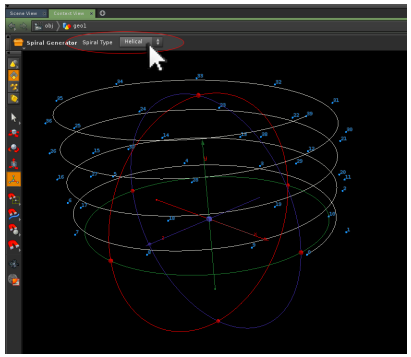


The channel reference of the Maximum segments parameter created by porting it up to the top level of the digital asset, can also be modified to subtract 1 from its value ensuring that a Number of Points value of 100 will result in exactly 100 points being generated for the polygon spiral (as opposed to 101 points).

INTERACTIVE VIEWER CONTROLS

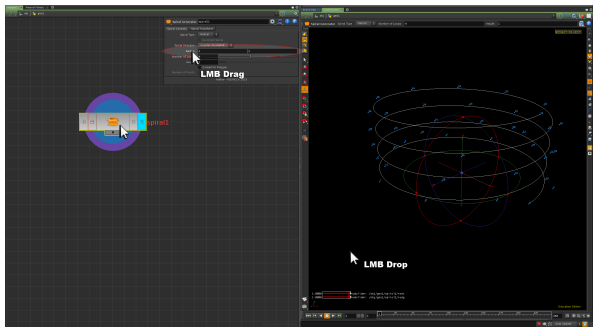
Set the Network Editor to look at the top level of the Spiral Generator Digital Asset. If **ENTER** is pressed with the **mouse over the Viewer**, the ported **Transform Handle** will appear. It can also be advantageous to have other core controls for the Spiral Generator appear in the Viewer.





To have core controls as part of the main Viewer interface, the desired **Parameter Description** can be modified to appear in the **Main & Tool Dialogs + Toolbar**.

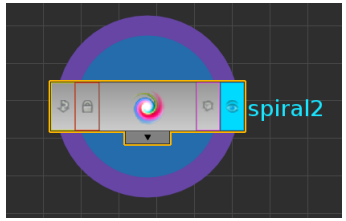
Parameters can also be **LMB dragged** directly onto the Viewer to create **Heads Up Displays** (**HUD Sliders**). These can also be activated as a permanent control for the Digital Asset if necessary.



See file spiral_generator_stage4.hipnc

EMBEDDING A NODE ICON, IMAGE OR FILE INTO A HDA

Under the **Basic** section of the **Edit Operator Type Properties** window it is possible to specify an icon for the Spiral Generator. All standard image formats will be accepted; however the icon must be square. A new Spiral Generator node will need to be created to see the icon after it has been created. By **default**, and **selected icon** will be **automatically embedded** into the HDA.



To **manually embed** other **additional external files** into a Digital Asset, the **file** must be loaded into the **Extra Files** section of the **Edit Operator Type Properties window**. It can then be called in a parameter using the **opdef:** command.

For example, an image called spiral.pic loaded into the Extra Files section of the Spiral Generator Digital Asset can be called as an icon in the Basic section by using the command:

opdef:/Sop/spiral?spiral.pic

This translates as:

opdef:/hda_level_location/hda_name?image_name.xxx

Embedded files can also be called in Digital Asset network parameters in a similar way.

WRITING THE DIGITAL ASSET HELP

Under the Help Section of the Operator Type Properties dialog enter the following text:

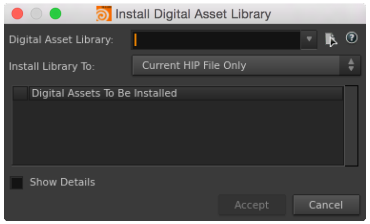
> **This is a Digital Asset for creating spirals. By default it will output a NURBS curve with a point number relative to the number of spiral loops.**

With the Help complete, Accept can be pressed in the Edit Operator Type Properties window.

NOTE: MMB Node help can be **assigned** using the **Edit Comments… option** located under the **Cog Button** of the **Digital Asset's parameter pane**.

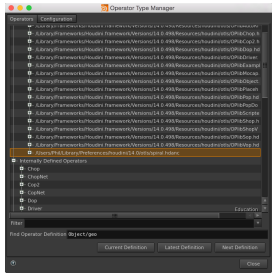
INSTALLING DIGITAL ASSETS

Digital Assets can be **installed** into any **Houdini scene** by going to the **main Assets menu** and choosing **Install Asset Library…** The Install Digital Asset Library window can then be utilised to locate and install an .otl file into Houdini.



MANAGING DIGITAL ASSETS

There is also an **Manage Asset Libraries…** option located under the **main Assets menu** that can switch between Digital Asset versions, and if necessary remove Digital Assets from the current scene.



REFRESHING DIGITAL ASSETS

Offline modifications to an **installed Digital Asset** can be **accessed** by going to the **main Assets Menu** and choosing **Refresh Asset Libraries**.

ADDING OTHER SPIRAL TYPES

Now that the **spiral HDA** is complete, **more spiral types** could be **added** by creating **other Point SOPs** in the correct location of the Digital Asset network. Some expressions to create other spiral types are:

Archimedean:

PosX	PosY	PosZ
$TX*$PT/$NPT	$TY	$TZ*$PT/$NPT

Logarithmic:

PosX	PosY	PosZ
$TX*pow($E,($PT/$NPT)*(1/(tan(10))))	$TY	$TZ*pow($E,($PT/$NPT)*(1/(tan(10))))

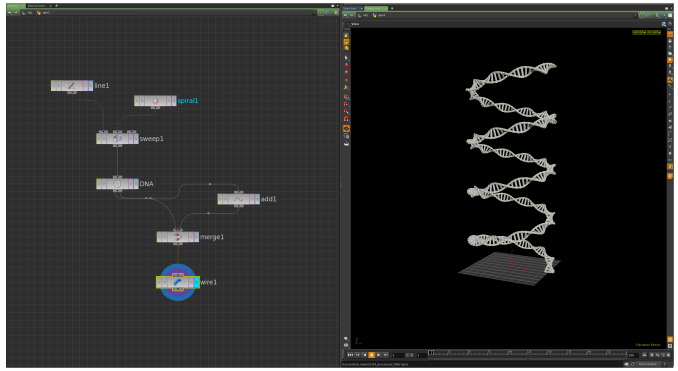
Involute:

PosX	PosY	PosZ
(cos($PT)+$PT*sin($PT/2))/$NPT	$TY	(sin($PT)-$PT*cos($PT/2))/$NPT

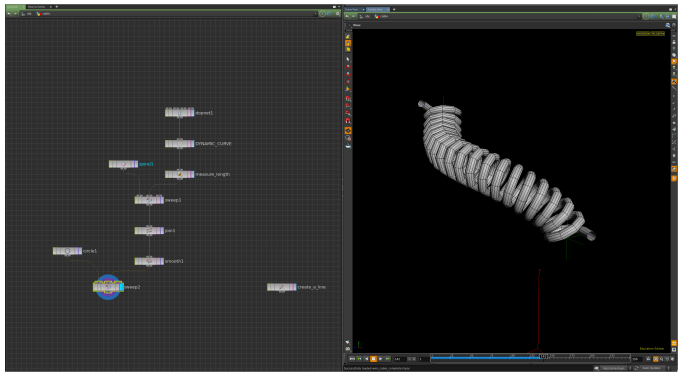
The advantage of creating a Digital Asset to manage the generation of spirals is simply that the expressions responsible for creating them never have to be seen again.

THE SPIRAL GENERATOR IN ACTION

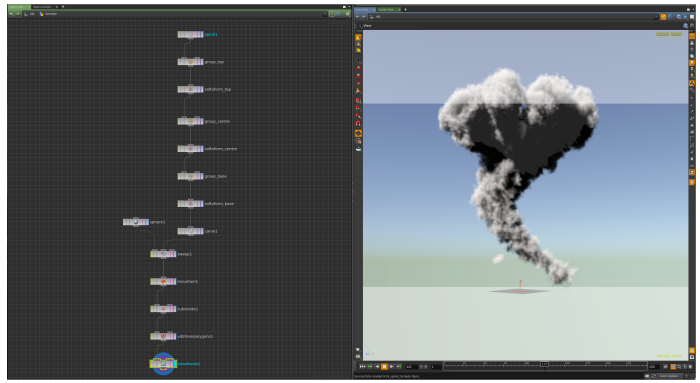
Open the scene **H14_procedural_DNA.hipnc**. This scene contains a DNA model which utilises the Spiral Generator as a construction node.



Open the scene **wire_cable_complete.hipnc**. This scene contains an elastic telephone cable that bends and stretches dynamically.



Open the scene **H14_spiral_tornado.hipnc**. This scene contains a simple tornado setup that generates a cloud volume from the Spiral Generator.



Any of these examples could eventually be turned into their own Digital Assets in their own right, with the Spiral Generator Digital Asset nested and referenced inside it.