Pipeline Consultants
rednineinfo@gmail.com

This is DRAFT Documentation for the Red9Studio tool pack. All comments, requests and feedback are welcomed.

For installation please follow the Install doc inside the modules themselves. Install is done via the Maya module systems to make the process more flexible.

Setting up the StudioPack:

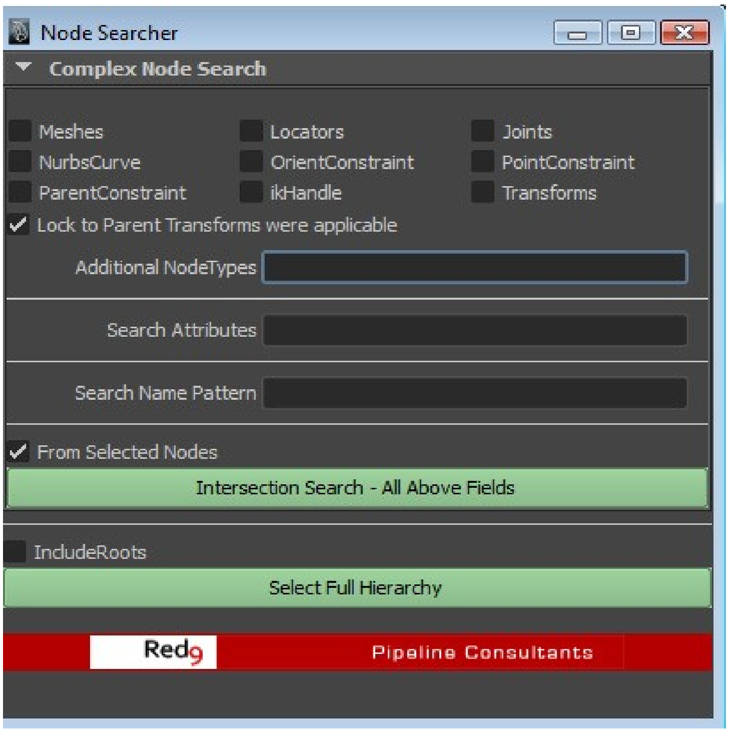
Please read the install instructions in the pack itself as these change as the code evolves. Once installed you should get a new menuItem in the main Window Red9 which will link you to all the UI's. The API behind this is also worth taking a look at as there are some very useful generic functions which you may find of use.

Thanks for taking the time to test this Pack

Red

NodeSearcher:

The NodeSearcher and the filterNode Api search code is the backbone of most of the studio code. The main functionality is an Intersection Searcher that runs 3 different filter function methods. These act together allowing you to find exact nodes either within complex hierarchy structures or at a Scene level. The 'From Selected nodes' checkbox controls the search level – Hierarchy or Scene



☐ Meshes ☐ Locators ☐ Joints
☐ NurbsCurve ☐ OrientConstraint ☐ PointConstraint
☐ ParentConstraint ☐ ikHandle ☐ Transforms
☑ Lock to Parent Transforms were applicable

Additional NodeTypes

Search Attributes

Search Name Pattern

☑ From Selected Nodes

☐ IncludeRoots

Most of the checkboxes are for nodeType searching, very similar to Maya's `listRelatives -type` call with one exception, it has handling to clamp the results to transform nodes where applicable. For example, nodeType 'mesh' would normally give you the mesh shapeNode back, but with the 'Lock to Parent Transform' the return will be the parent transform node above it. This is very useful for animation functions.

- **Additional NodeTypes:** This is a list, separated by ',' so you can build a filter of additional custom nodeTypes over and above those flagged by the checkboxes.
- **Search Attributes:** This will search nodes for a given attribute/attributes. The input is a list, separated by ',' allowing you to specify multiple search attrs.
- **Search Name Pattern:** Check node names that contain the given name strings. This accepts a list and is basically a Python regular expression, allowing you to clamp the search more accurately.

Name Pattern Examples:

> **^**Wrist will clamp the search to the start of name : here Wrist_Ctr = Match, L_Wrist_Ctr = No Match
> Ctr**$** will clamp to the end: here L_Wrist_Ctr = Match, L_Ctr_Wrist = No Match

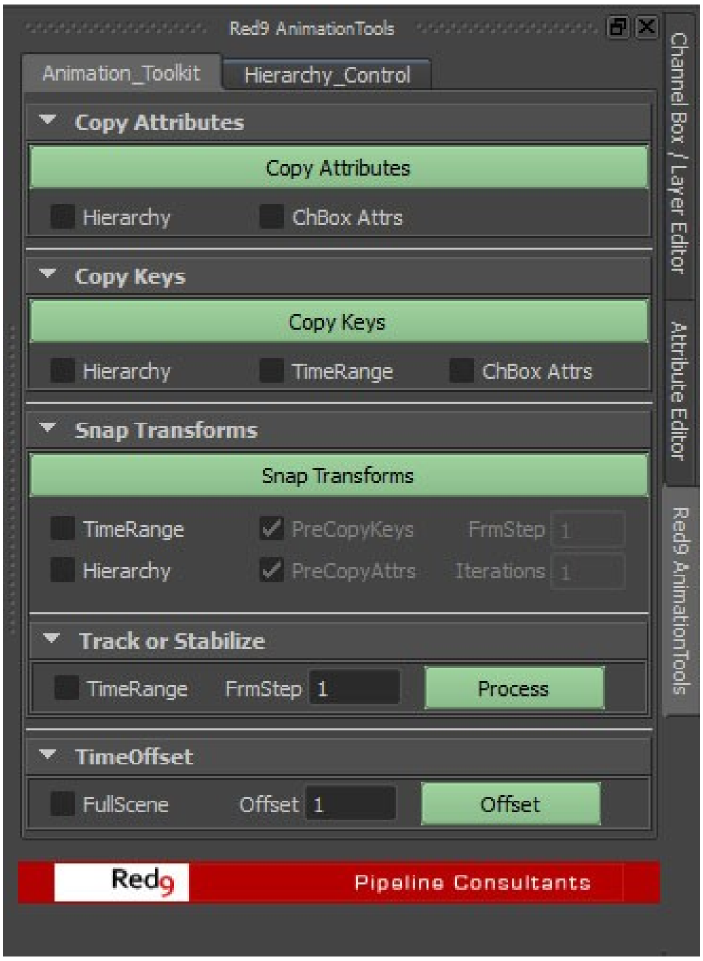
All 3 of these search methods are added together to form a single intersection search, only returning nodes for which all 3 are true. Blank fields are removed from the search.

Select Full Hierarchy:

This is a simple listRelatives(ad=True) wrapper with the exception that it allows you to specify that the rootNode be included in the return. Note: If a CharacterSet is selected as root, the this will return ALL member nodes of that Set.

AnimationTools:

This toolset is aimed to speed up animation workflow through simple yet powerful functionality. Many of these are tools which are sadly lacking in Maya by default, require custom scripts to achieve, or are implemented so basically that they're practically useless.



□ Hierarchy □ ChBox Attrs

□ Hierarchy □ TimeRange □ ChBox Attrs

□ TimeRange ☑ PreCopyKeys FrmStep 1

□ Hierarchy ☑ PreCopyAttrs Iterations 1

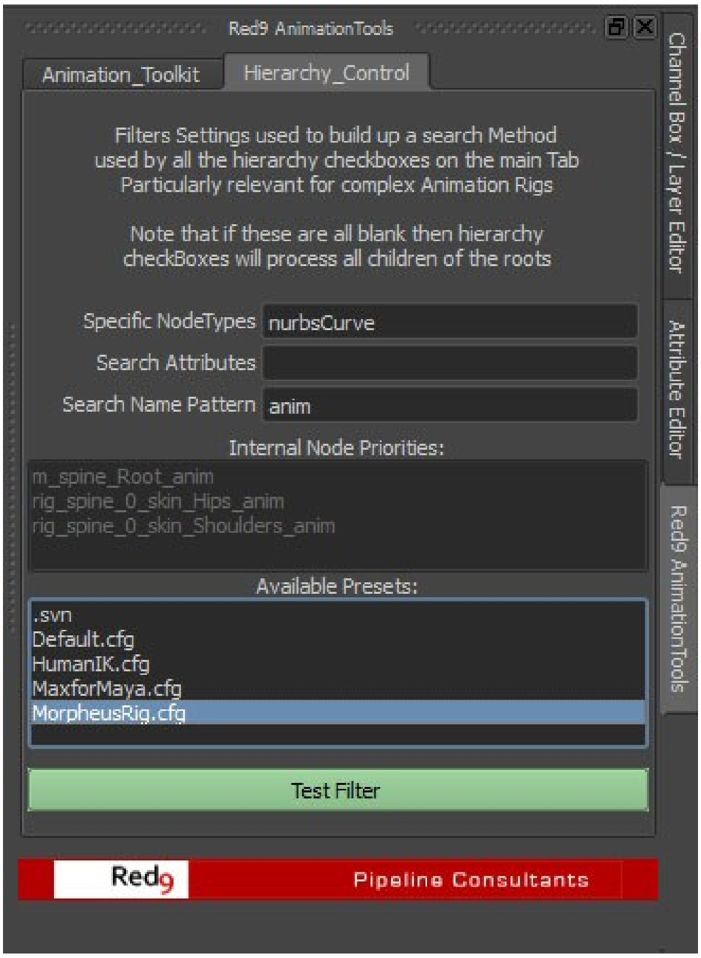
□ TimeRange FrmStep 1

□ FullScene Offset 1

The Animation UI currently supports 5 core functions although this is expanding rapidly:

- **Copy Attributes** - Copy Attribute data between selected nodes, hierarchies or Character Sets.
- **Copy Keys** - Copy Keys between selected nodes, hierarchies or Character Sets.
- **Snap Transforms** - Snap transform nodes together, either selected nodes, Character Set members or carefully filtered nodes using the FilterNode calls.
- **Track or Stabilize** – Either track objectA to objectB or stabilize objectA keeping it still regardless of animation inputs. This now supports Component level tracking.
- **TimeOffset** – Offset animation data, audio and clip times in one go. Either at scene level, or selected nodes.

Hierarchy Control:
This is new in the pack, it lets you carefully setup how all the hierarchy checkboxes in the main tab are going to deal with nodes under the selected roots. For complex animation rigs you don't want to be processing entire hierarchies due to the number of utility nodes and groups involved so this lets you setup what we're going to be processing. It's basically a way to ID only you're main animation controllers. I've added a number of presets to this which have been tweaked for better performance on certain widely available rigs. The bulk of this is the same as the NodeSearcher UI described above.



Filters Settings used to build up a search Method
used by all the hierarchy checkboxes on the main Tab
Particularly relevant for complex Animation Rigs

Note that if these are all blank then hierarchy
checkBoxes will process all children of the roots

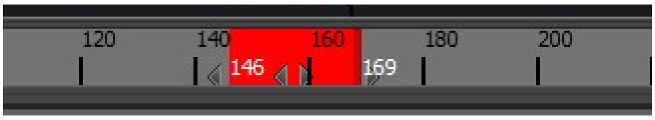
Specific NodeTypes: nurbsCurve
Search Attributes:
Search Name Pattern: anim

Internal Node Priorities:
m_spine_Root_anim
rig_spine_0_skin_Hips_anim
rig_spine_0_skin_Shoulders_anim

Available Presets:
.svn
Default.cfg
HumanIK.cfg
MaxforMaya.cfg
MorpheusRig.cfg

Selections: If you run either the Snap or Copy's without hierarchy they'll process data from nodes selected as pairs, so if you select in order objA, objB, objC, objD, objE, objF they'll process A>B, C>D, E>F in one go.

TimeRanges:
There's a consistency in the way we deal with time and selections across these which you need to be aware of. When processing TimeRanges a selected range is as below, shift, click drag in timeline.

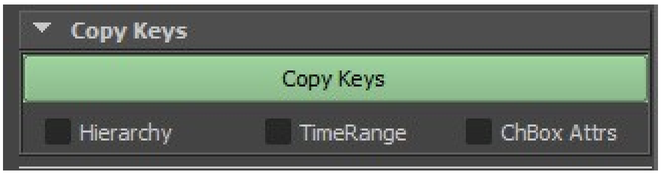


Timerange selected process f146-169

When the TimeRange's are checked the code looks to see if you have a selected timerange in the timeLine and if so, uses it (as above). If no range is selected then it uses the current playbackrange.

Copy Keys:

Copy all found keys between nodes. Selection can be multiple objects, 2 Character Sets or 2 root nodes for hierarchy filtering.



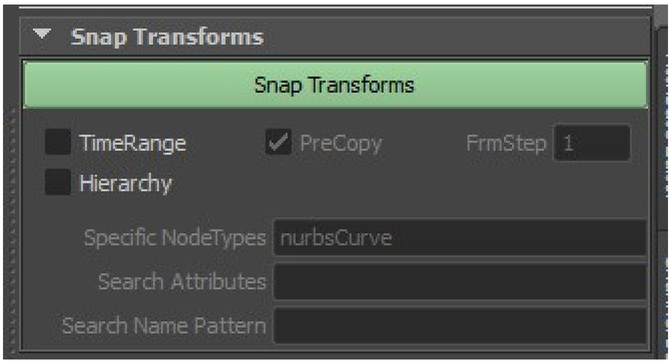
Hierarchy : With this off then the code will process selected pairs objA>objB, objC>objD, as described above. With it on it will filter both selected root nodes for all child transform / joint nodes, then run name matching code to build up a process list, before transferring the key data over. Unlike Maya's hierarchy copy this very carefully manages the node pairs to copy between. If the selected roots are both Character Sets then all members of these sets are processed.

TimeRange : As described above but with one exception, if OFF it will copy without any time restrictions, thus copying all keys over all time. With it ON it will process either a selected timeRange block, or the current playback timeline.

ChBox Attr : If this is on then it looks to see if any channels are selected in the channelBox, if so then it will limit the copy process to just those attributes.

Snap Transforms:

One crucial thing that's missing in Maya is a simple snap function, the ability to snap nodeA to nodeB, either statically or over time. This code fills that gap allowing you to not only snap nodes over time, but also to snap entire animation rigs and hierarchies over time.



☐ TimeRange ☑ PreCopy FrmStep 1
☐ Hierarchy

Specific NodeTypes: nurbsCurve
Search Attributes:
Search Name Pattern:

TimeRange : If this is OFF then snap only the current frame. If ON it will process either a selected timeRange block (see above) or if none is selected process the current playback timeline.

PreCopy : With timeRange enabled this becomes active. This runs the copy keys code above on all nodes prior to the snap itself. This is so that all secondary channels are transferred over at the same time. It also helps in the snap of complex heirarchies.

FrmStep : Steps in frames between each snap, allows you to key up blockouts at regular sample intervals.

Hierarchy : With this OFF the code will process selected pairs objA>objB, objC>objD, as described above. With it ON and 2 Character Sets selected it will process all members of each set, matching them by node name as it does so.

Turning this ON also turns on all the advanced hierarchy processing. This is the most complex to understand, although the principal and code is the same as the FilterUI calls. When snapping hierarchies you really need some idea of what to snap within those hierarchies, how to filter the children of the root nodes. The default is to filter for all nurbsCurves and snap those together. This is because normally nurbsCurves would be used as the controllers for rigs so it kind of makes sense.

For a more accurate snap you need to understand what the filter code does. This is the same as used in the FilterUI so it's worth understanding, or at least for somebody in the studio to understand the power of this filter core. There are 3 filters, which can be used together to accurately pin-point the required nodes.

Iteratrions: This is the number of cycles that the hierarchy list is processed. Because we have no idea of the order in which hierarchies should be snapped (especially in complex rigs) I've given you the ability to run each snap frame x number of times, given more accurate results but slower performance.

- **searchTypes**: this is a filter for nodeTypes, it wraps Maya's `listRelative(nodeType=xx)` but alters the order in which things are returned to better suite processing. In Maya all nodes are of a type that can be queried. Lets say you wanted to snap a joint hierarchy.... you'd put 'joint' into this field. This is also a list so you could put joint,nurbsCurve to find both joints and nurbsCurves.
  Note that this textField now has a fast RMB selector for common node types.
- **searchPattern**: this is a name based filter, looking for pattern matches on the node names themselves. Again this is a list so can have multiple entries. Eg: 'Wrist,Feet' = find only nodes who's name includes these. See NodeSearcher for more detail
- **searchAttrs**: this is the most powerful for riggers, it searches for specific attributes on the nodes, often you'll have markered controllers up with a custom attribute for other tools. This allows you to hook into this.

put these 3 together and you get filtering that enables you to find very specific nodes within a hierarchy.

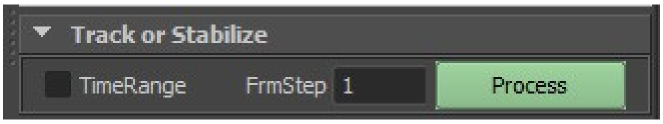
eg: filter for all joints who's name includes 'L_' , or filter for all nodes that have a marker attribute on them and are of type 'nurbsCurve'

TIP: If a single hierarchy snap fails to match everything, increase the number of iterations.

The order of the snap is crucial, and because this is a generic tool it has no idea of the actual parent spaces of any of the controllers. Ideally for a rig you want to snap the Root_Cntrl, then the Hips and Spine, then the Arms and Legs, snapping through the hierarchy in order. This is an ideal case, unfortunately because of parent switching in rigs that's not always the case. Also because this is hierarchy filtering it doesn't actually know what is what within the lists.

Track or Stabilize:

The general idea is to either lockdown or to track objects without building complex space switching hierarchies or constraints. The Process button runs on the current frame, then moves the timeline forward (or backwards) by the frmstp value. The code runs backwards (negative frmstep) as well as forwards. The timerange is as all other functions in the AnimationUI.



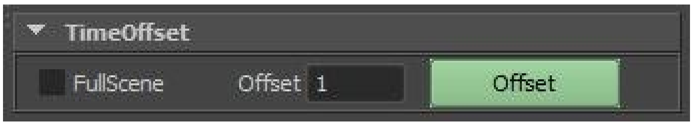
This button switches functionality depending on the number of objects selected.

STABILZE: If you only have a single object selected the code processes in STABILIZE mode. This will lock the object in place regardless of it's underlying animation, very good for locking or stabilizing complex MoCap data on controllers like feet where they have to plant firmly on the ground.

TRACKING: With 2 objects selected the code switches to TRACKING mode. Here the second selected node will track the first with the current offset maintained. This is great if you have 2 characters interacting and you want to firmly lock one controller onto a part of another character.

In future releases this will also process at a component level, allowing you to select a poly or vert for the snap reference.

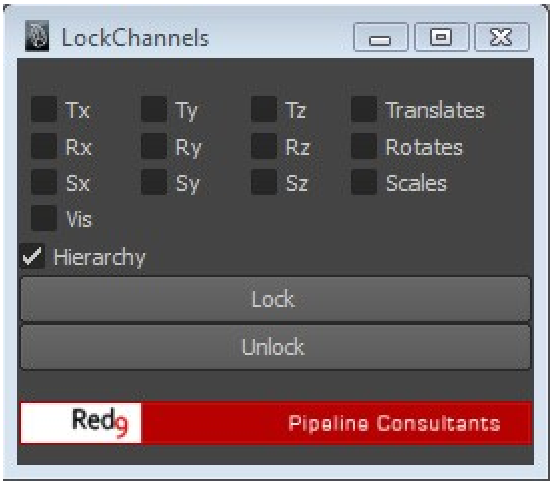
TimeOffset:



This is designed to manage scene time. If 'Scene' is checked then the code will Offset the entire scene, including the playback timelines, searching for animcurves, trax clips and sound nodes to offset. If not, then the code works on the currently selected nodes and the big thing here is that these can be anything in the scene, the code will automatically go find the relevant systems to offset. One issue with the Trax editor is you can't offset multiple audio nodes, or clips, but with this you can. If a Transform node is in the current selection then the code hunts through it's history to find relevant animCurves to offset.

LockChannels:

A useful yet simple way to manage standard channel visibility for hierarchies. Very useful for riggers. Hopefully this is self explanatory.



LockChannels

☐ Tx ☐ Ty ☐ Tz ☐ Translates
☐ Rx ☐ Ry ☐ Rz ☐ Rotates
☐ Sx ☐ Sy ☐ Sz ☐ Scales
☐ Vis
☑ Hierarchy