
Nebula Pipeline Toolbox

Talha Ahmed

Mar 31, 2020

CONTENTS

1	Introduction	1
1.1	What is Nebula Pipeline?	1
1.2	How to Read	2
1.3	Purpose and Audience	2
2	History	3
2.1	Legacy of ICE Animations	3
2.2	Influential Projects	3
2.3	The Pipeline Team	4
2.4	The Way Forward	4
3	Background	5
3.1	No Panacea!	5
3.2	The Production Pipeline	5
3.2.1	Observations	5
3.2.2	Deployment Architecture	9
3.2.3	Required Software	10
4	The Tool Box	11
4.1	Production Planning	11
4.1.1	Episode Planner	11
4.1.2	Sequence Planner	13
4.2	Assets Development	15
4.2.1	Assets Explorer	16
4.2.2	Remap Textures	20
4.2.3	Shader Transfer	21
4.2.4	Proxy Cache Switch	23
4.3	Layout and Animation Tools	24
4.3.1	Create Layout	24
4.3.2	Multishot Export	27
4.3.3	Moctor	30
4.4	Lighting and Rendering Tools	32
4.4.1	Add Assets	33
4.4.2	Setup Master Scene	33
4.4.3	Create Shots	36
4.4.4	Matte IDs	39

4.4.5	Redshift AOV Tools	40
4.4.6	Pre-CC	41
4.4.7	Scene Bundle	43
4.5	Post Production	44
4.5.1	Backdrop Tool	44
5	The Last Word	49

INTRODUCTION

This document attempts to give an overview of the Nebula Pipeline and some of the functionality it offers.

In order to get an overview of the toolbox you can also watch the following youtube video



Nebula Pipeline Introduction

1.1 What is Nebula Pipeline?

Nebula Pipeline is a collection of tools that deploy at different stages of the pipeline in order to effectively manage a CG production by the use of a central database.

The pipeline has evolved from automation attempts done in production. Therefore, The advantage that these tools offers is more due to the fact that it has been used to produced many hours of animated content across artists of various roles and skillset.

Due to the nature of the projects it has been developed to support it is most well-gearred towards supporting episodic animations while it has also been used effectively in creation of shorts and VFX features.

1.2 How to Read

This document briefly describe the various tools that make up the Nebula Pipeline.

In order to put all the tools in context the *Background* chapter gives an overall view of the pipeline that the scripts can be used together to implement.

The next chapter, *The Tool Box*, goes on to describe each tool briefly. Each description is aided with screen shots of the UI alongwith a listing of its feature and small description of its usage along with the required software. Each tool description is placed within the context of its use in the production process so a fair idea can be developed of the value it offers.

The chapter *History* is more like a behind the scenes view of how the pipeline came into being. It could make for interesting knowledge of those who want to have a view of the rather small Pakistan CGI production industry.

1.3 Purpose and Audience

Happy Reading!

—The Author

This document is intended to give an introductory insight to someone who would like to employ the pipeline tools at their studio. However, the brief usage descriptions can also be used by its users to get an idea about the normal operation of each tool.

HISTORY

2.1 Legacy of ICE Animations

ICE Animations has been an important contributor to the local animations and VFX industry in Pakistan in a variety of ways. It is one of the few institutions that has actively spent its time and resources in experimenting with growth models.

For example, it has held the reputation of being the place where young aspirants in CG get professional training and their first exposure with local and international clients. Many have moved on to creating their own businesses and on to jobs in international studios.

Similarly, ICE Animations was also the only local studio to train and support a number computer engineers working on various aspects of CG. The pipeline team was also a part of such experiments laid out to improve quality and velocity of production.

2.2 Influential Projects

On the outset the financial and professional motivation of working in CG is towards the movie and gaming industry in the west, and there have been some VFX features executed locally that address that motivation. But, technically some of the projects that originated closer have been more Influential.

Bankay Mian was a TV show of temporary local fame because it addressed politics via a comical cartoon character, however, there were some technical achievements behind it. Firstly, it involved use of a state of the art motion capture system at that time. However, its pipeline was an important piece of the puzzle. The repetitive and limited scope of the production meant that automation could deliver many minutes of quality content per day with limited computing and human resources.

Mansour was a emirati TV show which required use of same assets over many episodes and several seasons. As more episodic animations project were acquired, the need for a production pipeline emerged to solve the recurring issues. This was a classic opportunity for developing a pipeline and remained the main motivation behind building of a new team.

2.3 The Pipeline Team

Back in 2012, A new team of raw computer programmers was given the mandate for getting trained in CG software and aimed to address production issues using *agile* development practices.

The pipeline team started with delivering tools for workflow and automation, addressing specific intra departmental issues. Later the effort focused on creating systems and addressing the pipeline as a whole by arranging tool menus and developing a digital asset management system.

The nature of software projects is such that they are perpetually under development. It requires experimentation and working through failures. So, it is during the course of the last three seasons of the *Mansour* and other similar projects a pipeline evolved which had a noticeable positive impact in the effective utilization of resources and helping artists focus on creativity.

2.4 The Way Forward

The local industry in Pakistan has now moved to newer model where the core teams are small and the type and size of projects has either remained unchanged or increased, and there is more reliance on project-based contracts and freelance artists.

While, in the current scenario the need for a permanent team to look after the pipeline has increased, the studios are unable to afford such a team on their own.

One solution could be the model of the electric company where one dedicated team can be used to address the issues faced by many different studios and production teams, and build on the work done over the years. The Nebula Pipeline initiative is an effort that hopes to address this gap of needs with this newer business model.

BACKGROUND

Each tool contributes to a specific stage in the pipeline and some of them are dependant upon the design assumptions and output of the other. Therefore, it is necessary to have an overview of the production pipeline and the context for their development.

This is intended to give an insight to the reader of how the various tools in the pipeline depend on each other and can function together to deliver a production.

3.1 No Panacea!

There can be many ways of solving a production into a pipeline, while no one solution is good enough for everyone. The effectiveness of the structure of a pipeline depends upon the business model and culture of the organization, its size and the history of the type of projects done under it.

With these factors in mind we shall now have a look at the proposed pipeline.

3.2 The Production Pipeline

The underlying pipeline can be schematically represented by the following diagram.

3.2.1 Observations

Now we shall have a look at some important observations about the pipeline view that this figure represents.

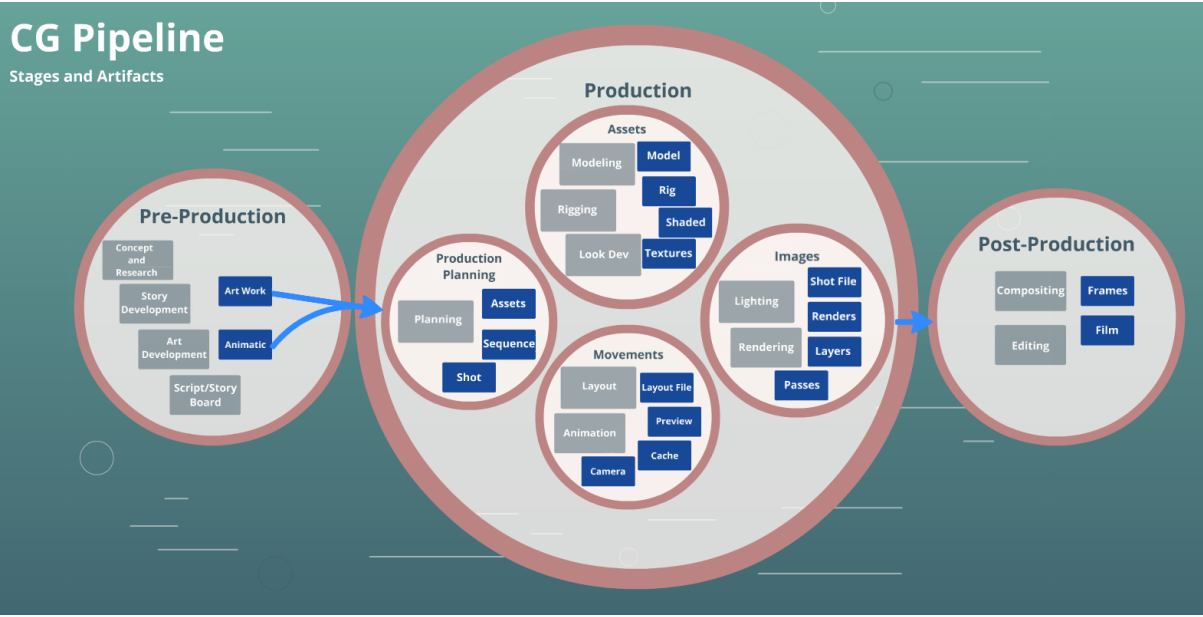


Fig. 1: Schematic View of the production pipeline
Each phase of this image is represented by a circle. The blue boxes represent the artifacts produced in the phase and the grey boxes represent the processes involved to produce them.

Symbol	Meaning
	Pipeline Phase
	Pipeline Process
	Artifacts

Business Model

The image shows the division of the complete production process into three main areas. Pre-production, production and post-production. Moreover, the assets, and production planning have been placed under the *production* area. This is due to the nature of the projects done at many studios who integrate into the industry as a production service company. This means that the pre-production is done separately either outside the studio or in an art department which employs the services of the production department to get the project executed.

Considering asset development process as a part of production is very useful in this business model. Even in the case when asset development is carried out upstream, there is always some sort of prepping to be done for production. Securing your assets is a good exercise of this business model and helps the standardize the transition into production.

Medium Size Studio

The production pipeline that the current batch of scripts supports have been developed for a medium-sized studio.

This meant that teams were organized into departments according to their function in the production process. Each department has a supervisor who is responsible for seeing through their portion of the production with the limited hands at his disposal. One departments output is the input of the other and standardization is needed. This also presents a requirement for departmental review. Such a studio is required to develop a pipeline which can be used over several projects of often similar nature.

The second observation of note is that the assets processes are placed separately from the *movement* and *imaging* section of the production. This actively facilitates an episodic production where the assets may be finalized once and may be used repetitively inside episodes after relatively.

Type of Projects

Table 1: Types of Projects

Type	Animation	Live Action	VFX		
Short	✓		✓		
Episode	✓				
Feature			✓		

Animation vs VFX and Live Action

VFX projects are different from animation projects because of their requirement for photorealism. While photorealism brings a certain standard for quality into play it also makes production complex because it is still difficult to achieve.

Production wise it is different because of the dependance on *plates* i.e. the footage from shooting of live actors or environments. This shifts the focus towards post production as the goal is convincing integration of the plates with the VFX. This means that much of the work is dictated by the look of the final blended frame and many assets may just be 2-Dimensional in nature. This entails a need for fluidity towards the end of production in the hands of the *compositor* while any rigidity introduced by 3-Dimensionally synthesized assets should primarily cater for that need.

Episodic vs Feature and Shorts

Although animation productions whether Episodic or Feature length will have the same type of skill required from its artists but their production cycles especially related to asset development will differ.

Episodic projects will have assets created once per many episodes or production cycles. Most assets are produced without the knowledge of the role in the script and many times they are used again without many changes across episodes. Moreover, many versions of the same asset may be required in different productions in parallel. While this scenario might be a source for complications, it also provides opportunities for more reuse and thus increased efficiency in utilization of the asset team.

Assets in Shorts and Features on the other hand maybe tinkered with deep into production. Many assets, especially sets and environments, may only be used in once or sequence and thus may constantly require improvements even after being brought into productions, blurring the boundaries between assets development and production processes. This is the reason why we may not find a lot of emphasis on this distinction with shops who are focused on features.

Nebula Pipeline gives importance to the boundaries between asset development and production as it provides an opportunity to formalize handing over of tasks downstream. It also recognizes the requirement for supervisor review and approvals these points.

Timely reviews and approvals will limit redoes to a small stage in pipeline and result in a more efficient production cycle even in a feature scenario. This might mean that an asset becomes less modifiable down stream in production, and thus dictates the need to finalize them before being used in production.

Environmental Effects vs Character Effects

Effects that introduce real complexity into a production are the 3D effects that involve physical simulation. To deliver these convincingly requires focused expertise and capable hardware, this is the reason they are expensive.

The software that are required to do these tasks integrate into the pipeline through use of common formats such as *fbx* and *alembic*, and thus make the task trivial on the outset. Moreover, the small number of jobs seen alongwith the involvement of the major amount of effort to execute the task itself, it is difficult to lockdown a common strategy of dealing with it.

Character Effects such as cloth and hair simulation on the other hand are recurrent and more fundamental to the asset development process and the production pipeline. The pipeline needs to adapt to the requirements of such effects across all asset and production phases. The tools currently leave space for character effects by allowing multiple geometry sets and standardizing exists only for hand-animated geometry. However, there is room for automation here if such protocols can be agreed upon.

3.2.2 Deployment Architecture

The pipeline deployment comes between the *users* and the *Network Attached Storage*. The degree of completeness of the pipeline is inversely related to the number of times users need to manually access the data in the *Network Attached Storage*

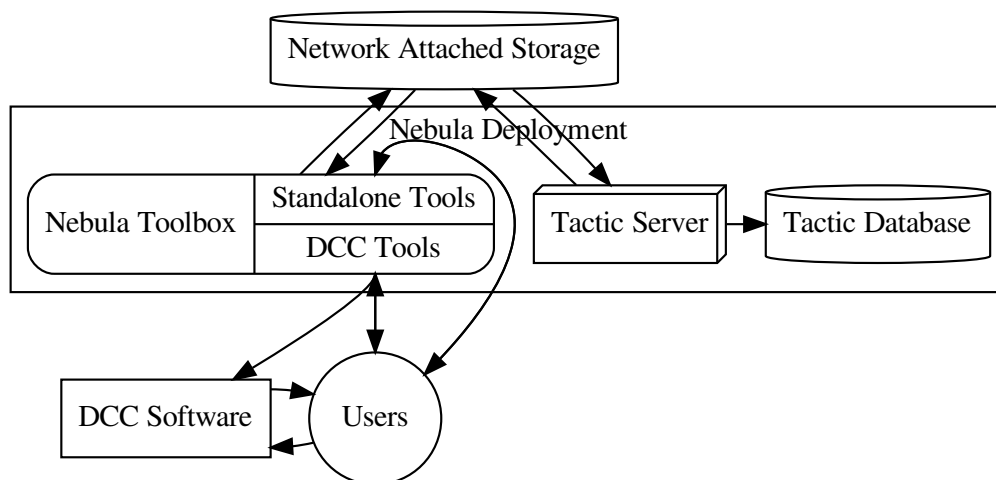


Fig. 2: Deployment Architecture

The Nebula Pipeline makes use of the *Tactic CMS* to manage its database and to provide production tracking.

3.2.3 Required Software

The pipeline tools require the following software to run

- Windows OS
- Autodesk Maya
- Redshift 3D
- The Foundry Nuke
- The Foundry Hiero
- Thinkbox Deadline
- Southpaw Tactic Open Source
- FFMPEG
- Image Magick

THE TOOL BOX

The descriptions of the tools has been arranged to be grouped under the stages of the pipeline they address. This can be useful in getting a context for the problem that the tool addresses during production.

4.1 Production Planning

4.1.1 Episode Planner

Episode Planner is a *Production Planning* tool used for population of episodes from a folder whose hierarchy describes the various shots and sequences of an episode.

Features

- Populate the tactic database with sequence, shots and frame ranges from scanning the filesystem

Screen Shot

Usage

Note: This tool assumes a specific hierarchy which can be achieved from use of templates in conforming software such as *Hiero* or *Nuke studio*.

- Cut the animatic into shots and sequences, and perform an export operation.
- Specify the *project* and *episode* from the drop-down list.`
- Provides the output path from the conforming software
- Hit *Populate*

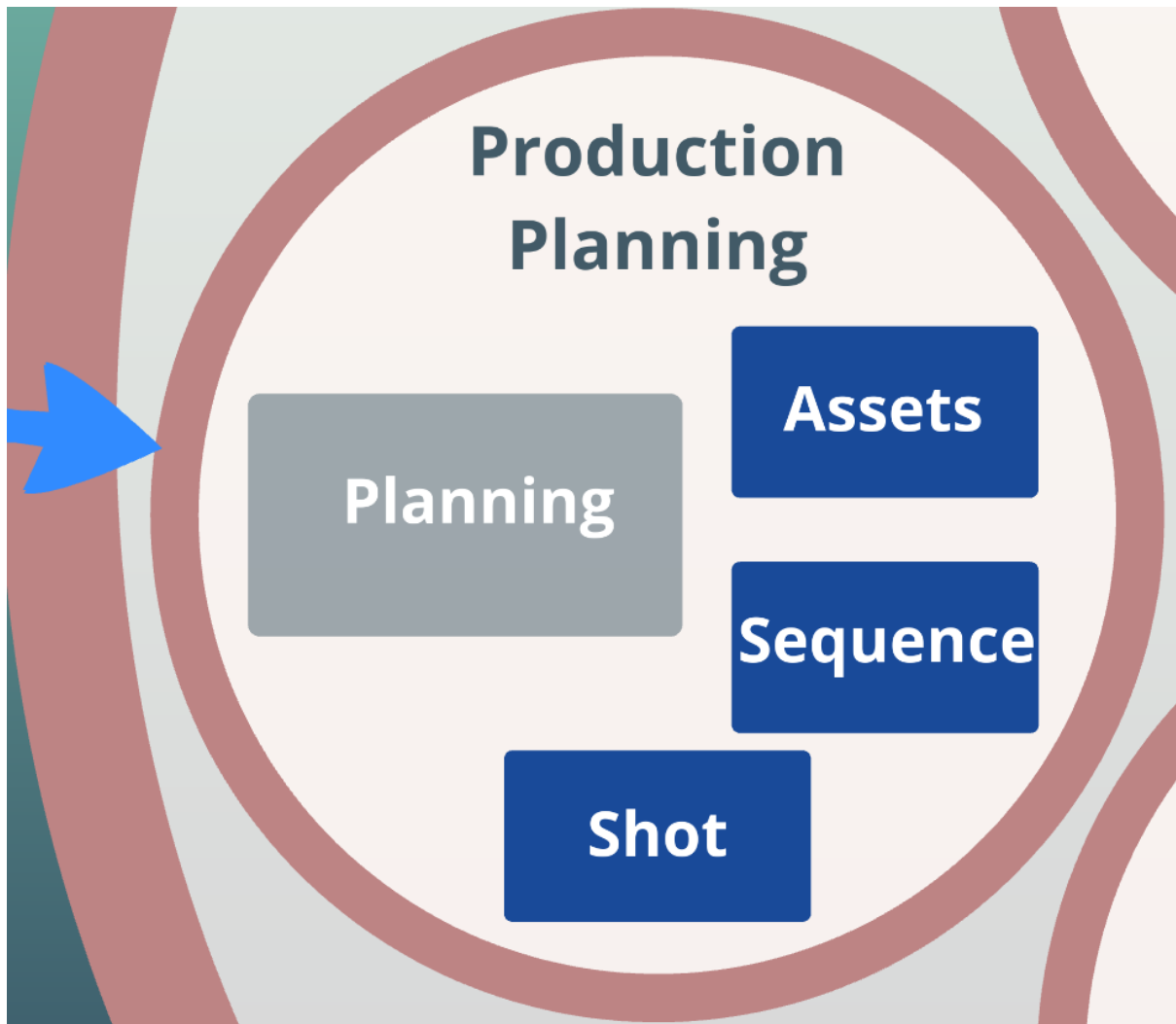


Fig. 1: Planning Phase

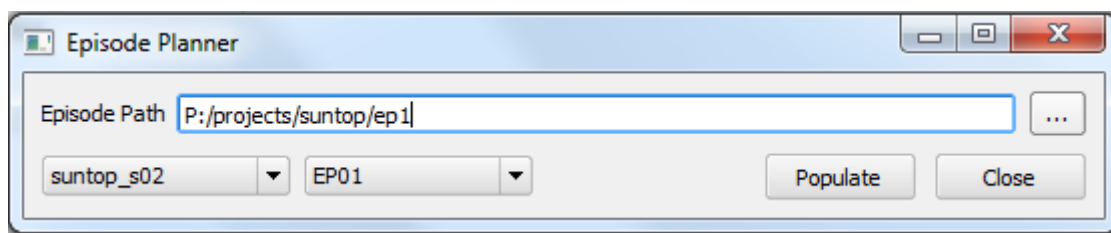


Fig. 2: Episode Planner Screen Shot

Warning: Do not perform this operation after the episode has gone into production. This may make you lose data related to production.

Tip: Use this tool only when the episode is being initiated.

Requirements:

- *Windows OS*
- *Southpaw Tactic*

4.1.2 Sequence Planner

Sequence Planner is an *Production Planning* tool that runs as a standalone application that helps to associate assets to sequences. This is a necessary step to perform before creating layouts using the *Create Layout* tool.

Screen Shot

Features

- List all the *assets* and *sequences* related to an episode.
- Add the required assets under the relevant sequences

Usage

1. Select *project* and *episode* from the drop down list on top to populate *assets* and *sequences* the top and bottom boxes.
2. Select a number of assets from the list above and click the + button on the relevant sequence to add the assets to the sequence. The changes are stored in the database immediately.

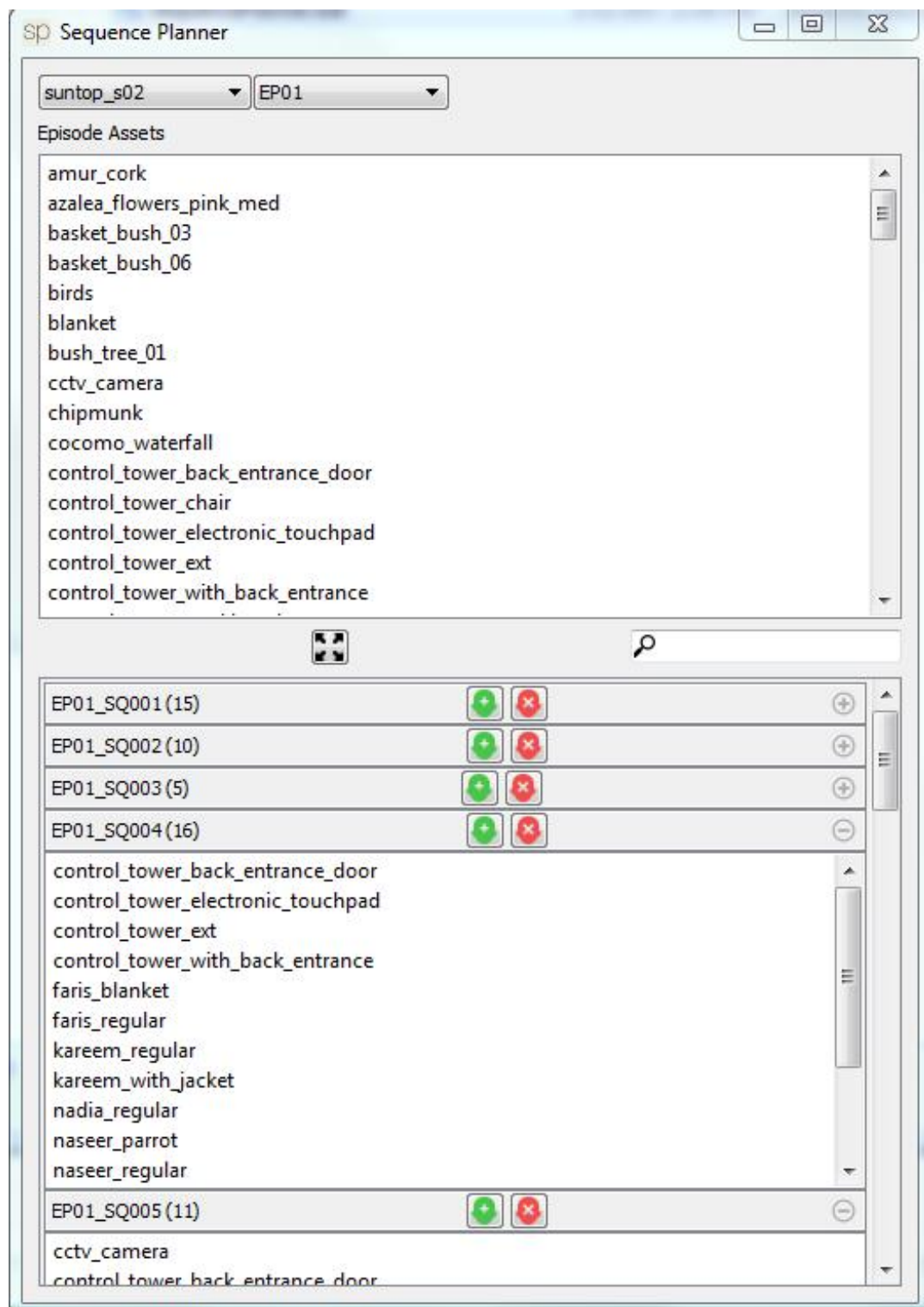


Fig. 3: Sequence Planner UI

Requirements

- *Southpaw Tactic*
- *Windows*

4.2 Assets Development

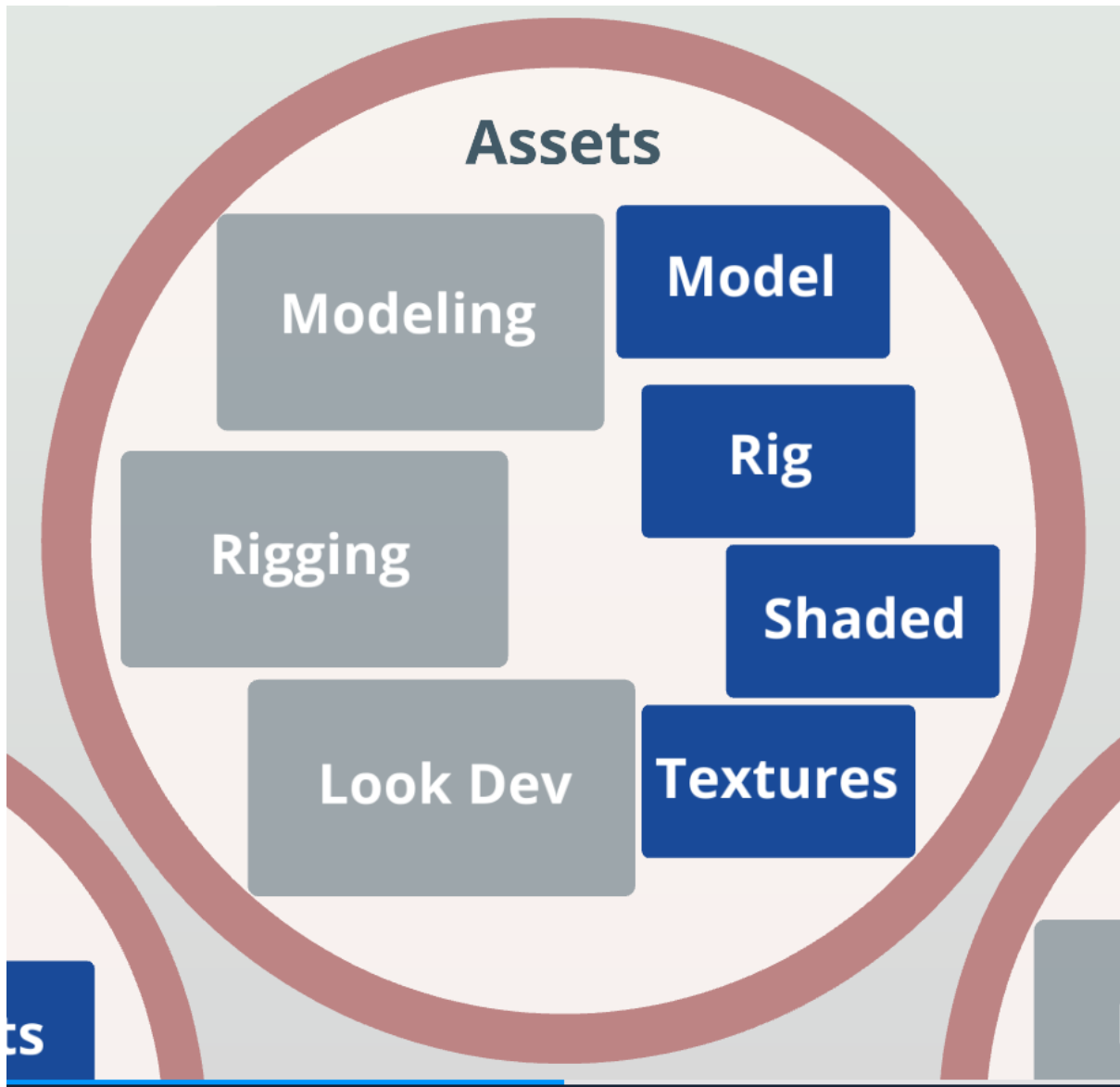


Fig. 4: Assets Phase

4.2.1 Assets Explorer

Asset Explorer is a comprehensive *Assets Development* tool that can be used to Create, Read, Update and Publish Assets.

The Asset Explorer is intended to run inside of *Autodesk Maya* but it can also run on *Microsoft Windows* as a standalone application.

Screen Shot

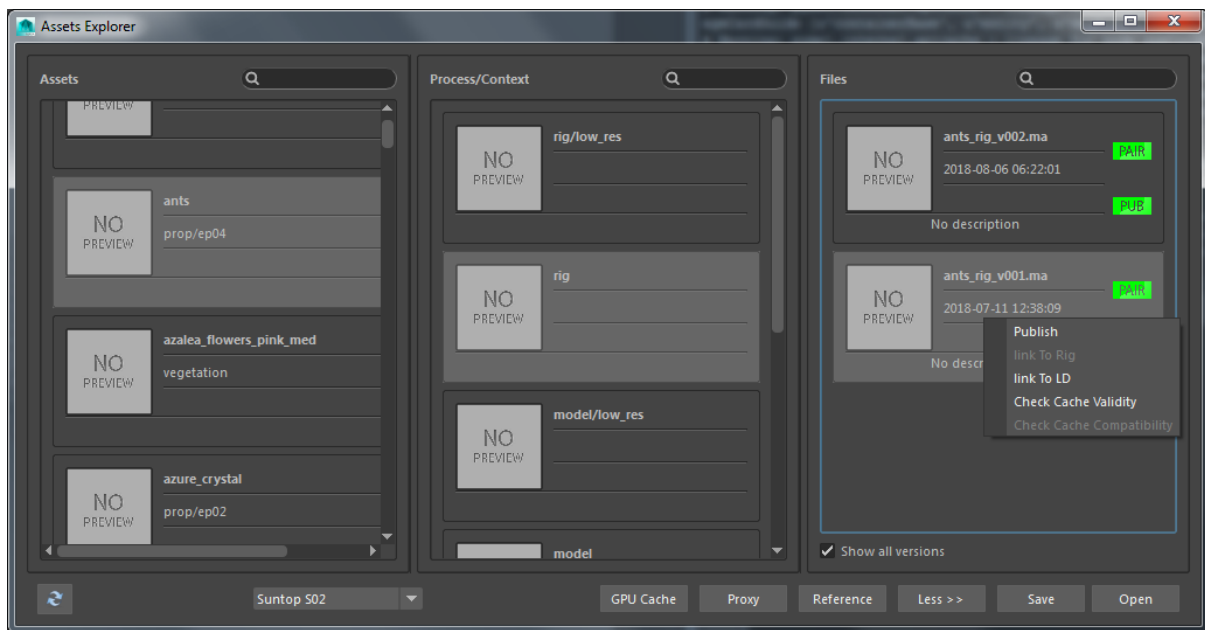


Fig. 5: The Asset Explorer UI running inside of maya

Features

- List all assets, contexts and versions related to a project.
- Import, Reference or CheckOut any asset version along with dependencies.
- Save assets to the *Tactic* server along with texture.
- Pair assets which quality for cache compatibility.
- Publish assets to *episodes* etc. for use in production.

Usage

The typical workflow of the asset explorer can be as follows:

- The user selects a project from the listing to get a list of assets from the tactic server.
- Filter the assets using the search box on the top of the left most column
- Select the context of the asset you need to work with such as model, rig or shaded to show the various versions published to the context.
- Select a version and hit the *import* or *reference* button to import or reference the required version.
- While a context is selected you can use the *save* button to perform *Asset Saving*.
- Right Clicking on a version will reveal the context menu which can be used to either perform *Asset Pairing* or *Asset Publishing*

Asset Saving

When the save button is pressed this reveals the Asset Check In dialog which can be used to create a new asset version in the selected context from the current maya scene.

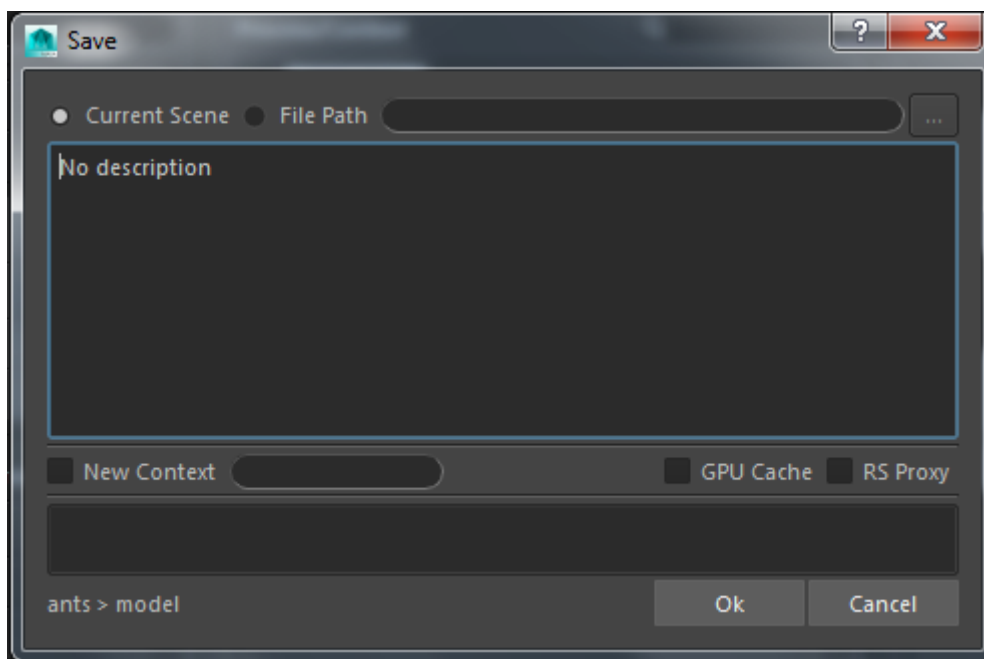


Fig. 6: The Check In dialog that allows you to save the current maya scene file to *Tactic*

Optionally the user can select geometry from the open scene to export as a *Redshift* proxy or a GPU cache (alembic) file.

The following behaviours are also associated with file saving:

- While saving a file it must contain a *geo_set* which is a maya set containing geometry nodes which must have a suffix of *geo* in its name. The absence of such a set or a

presence of an empty set renders the scene invalid and the asset explorer will show an error dialog.

- While saving files in the *shaded* context all the textures referenced in the current scene are also uploaded to the tactic server in the assets *texture* with their paths being updated to reflect their new location on the server.

Asset Pairing

Asset Pairing is the process of finding out if a particular version of asset rig file is cache compatible with a certain version of asset model or shaded file.

Once an asset pairing is performed this information is stored into the tactic database and the asset explorer can then be used to publish the asset to an episode in production.

Note: Asset pairing is not required and unavailable for *environment* and *vegetation asset categories*

Asset Publishing

Asset Publishing is the process of saving a version of an asset for use in a particular element of production most typically an *episode*.

Note: When an environment is published all the GPU and proxy references containing are also published along with it.

When an asset is published all the files related to that versions are copied to the directory of the particular episode and a versionless current version is made available for use in production.

Note: When a shaded model is published all its referenced textures are published along with it

Requirements

- *Windows OS*
- *Autodesk Maya*
- *Southpaw Tactic*

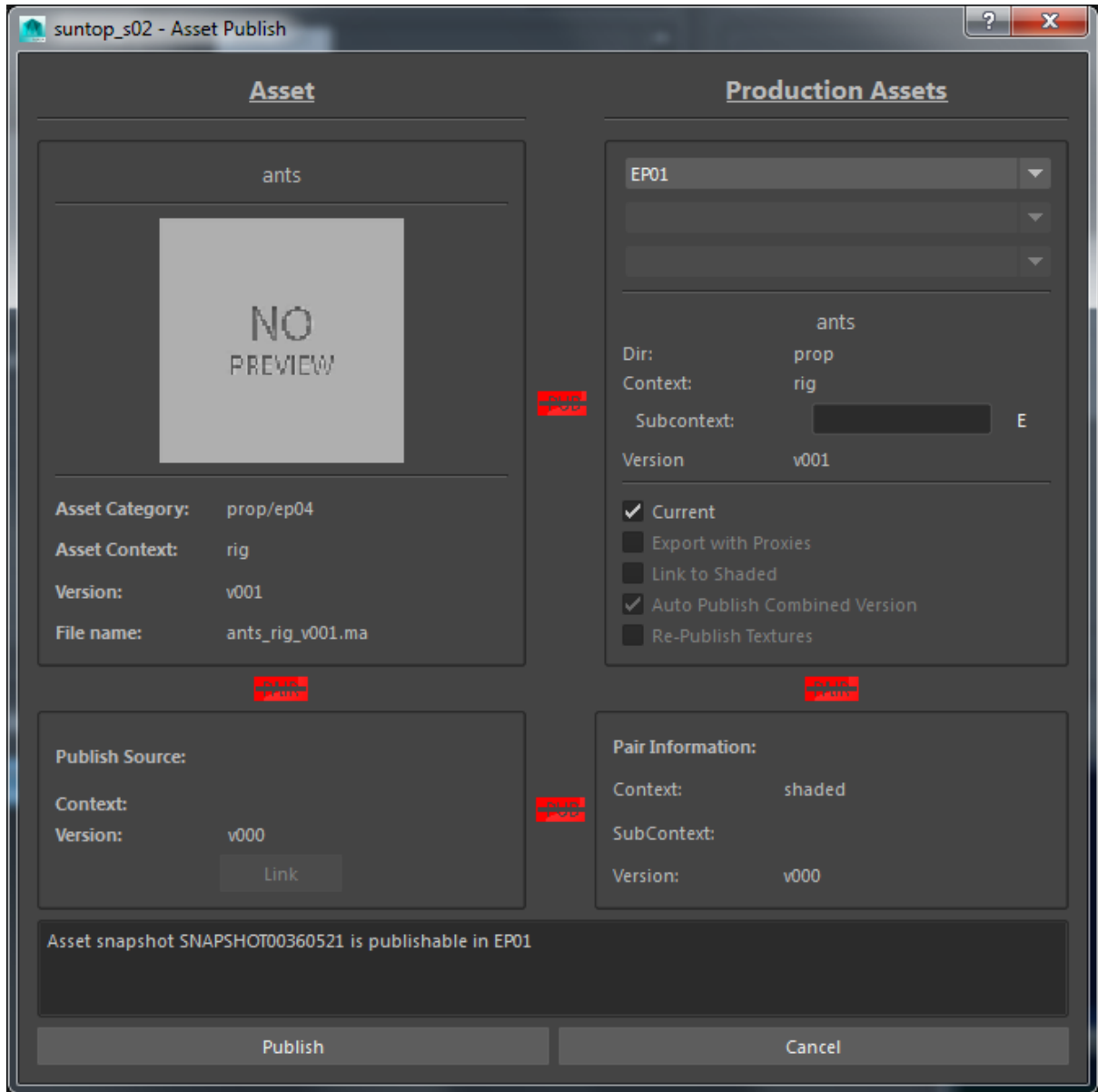


Fig. 7: A publishing interface showing if the current asset is publishable in the selected episode

4.2.2 Remap Textures

Remap Textures is an *Assets Development* tool that runs inside of *Maya* that helps artists to adjust paths of texture nodes after they have been relocated to a new path.

Screen Shot

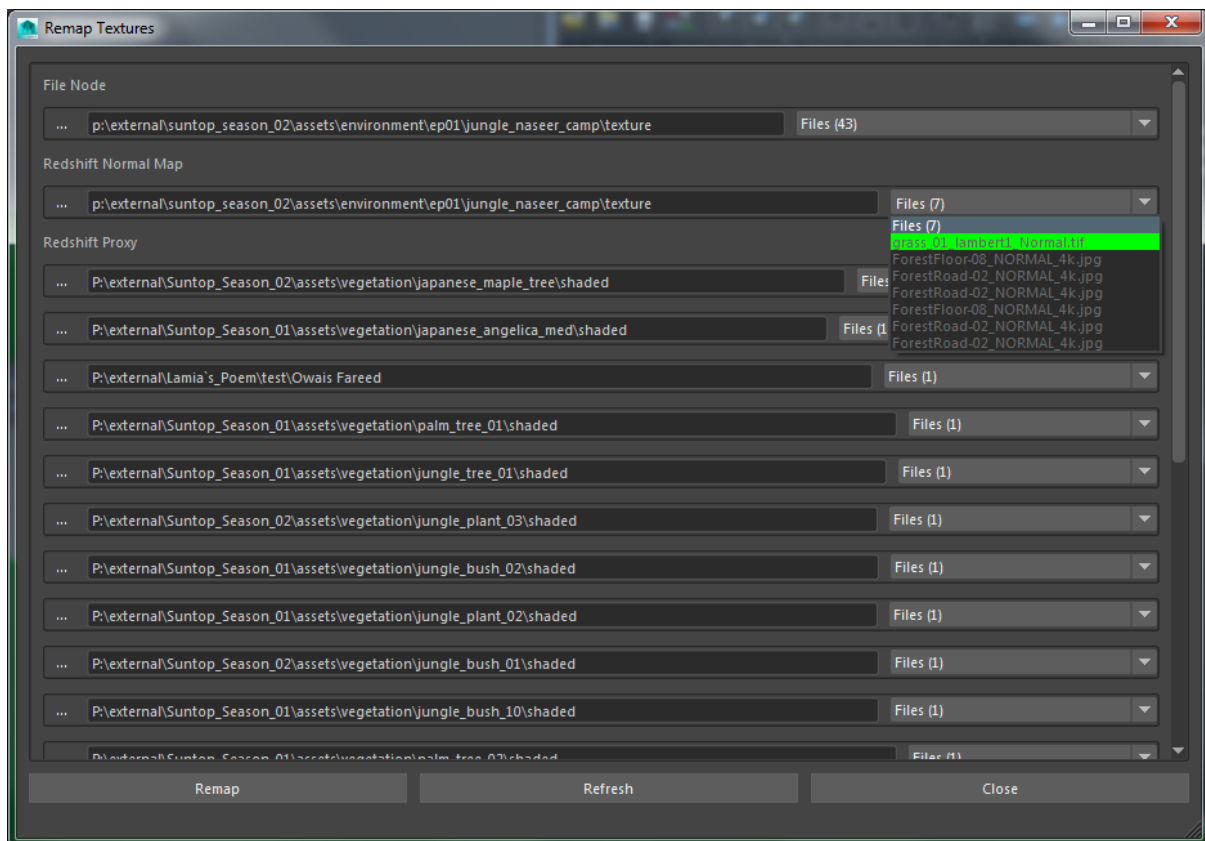


Fig. 8: Remap Textures running inside Maya.

Features

- List all the paths in the scene referring to texture nodes.
- List all the paths in the scene referring to *redshift* proxy nodes.
- Highlight the missing texture from the paths
- Change the path of all the textures to indicate change of location.

Usage

1. Make sure to hit *Refresh* to show status of the current scene.
2. The missing texture paths are highlighted.
3. Provide the new folder locations for the path you want to modify.
4. Hit `remap` to change the paths on the nodes.

Requirements

- *Redshift 3D*
- *Autodesk Maya*

4.2.3 Shader Transfer

Shader Transfer is an *Assets Development* tool that runs inside of *Maya* which helps in transferring shaders from one maya object to another.

Screen Shot

Features

- Transfer shaders and UV sets from single mesh to multiple single mesh
- Transfer shaders and UV sets from members of a set to members of multiple sets.

Usage

1. Select *Transfer Policy* from the drop down list either `Single to Single` or `Set to Set`.
2. Select the *source* mesh or set and hit `Add Source` button.
3. Select the *target* meshes or sets and hit `Add Targets` button.
4. Check the `Transfer UVs` button if required.
5. Hit `Transfer` to start the process.

Note: The number of faces on the source and target meshes should be the same if the shaders are assigned to individual faces.



Fig. 9: Shader Transfer UI from Inside of *Maya*

Requirements

- *Autodesk Maya*

4.2.4 Proxy Cache Switch

Proxy Cache Switch is a generic production tool that runs inside of *Maya* that helps users in switching from *proxies* and *gpu caches* or changing them from *hi-res* to *low-res*.

Screen Shot

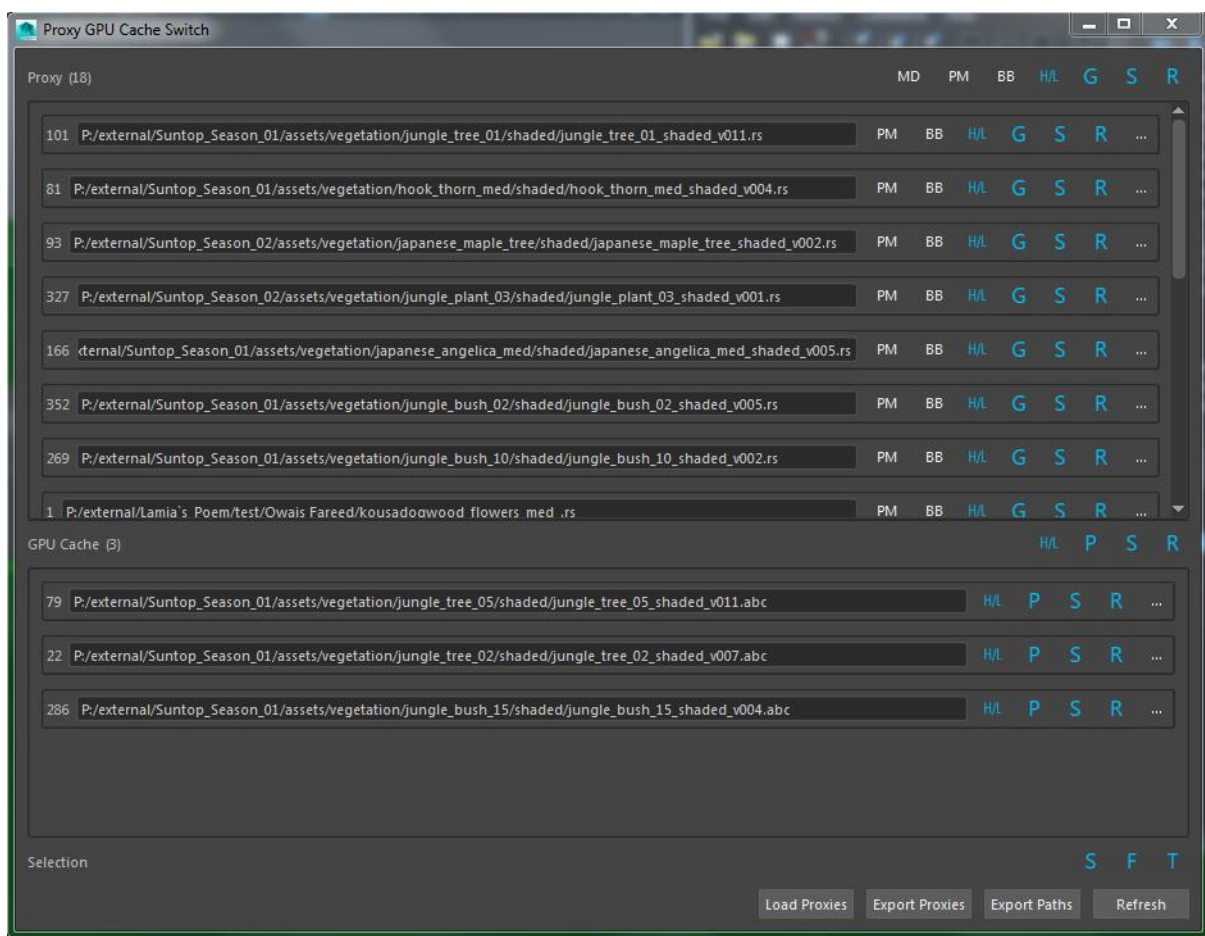


Fig. 10: Proxy Cache Switch running inside maya.

Features

- Switch between *proxy* and *gpu-cache* versions of the same object.
- Switch between *hi-res* and *low-res* versions of the same object.
- Change properties on *proxy* and *cache* nodes.
- Export proxies and paths from the scene.

Usage

- In order to switch from *gpu-cache* from *proxy*, hit the G button in front of it.
- To switch from *proxy* to *gpu-cache* hit the P button.
- To switch between *hi-res* and *low-res*, hit the H/L button.

Requirements

- *Autodesk Maya*

Asset Categories

Assets can be classified into the following *Asset Categories*:

1. Characters
2. Environments
3. Props
4. Vehicles
5. Vegetation

From the above asset categories only Characters, Props and Vehicles require a *rig* for animations while Environment and Vegetation only require *model* and *shaded* contexts.

4.3 Layout and Animation Tools

4.3.1 Create Layout

Create Layout is an *Animation* tool used for initiating animations the sequence level.

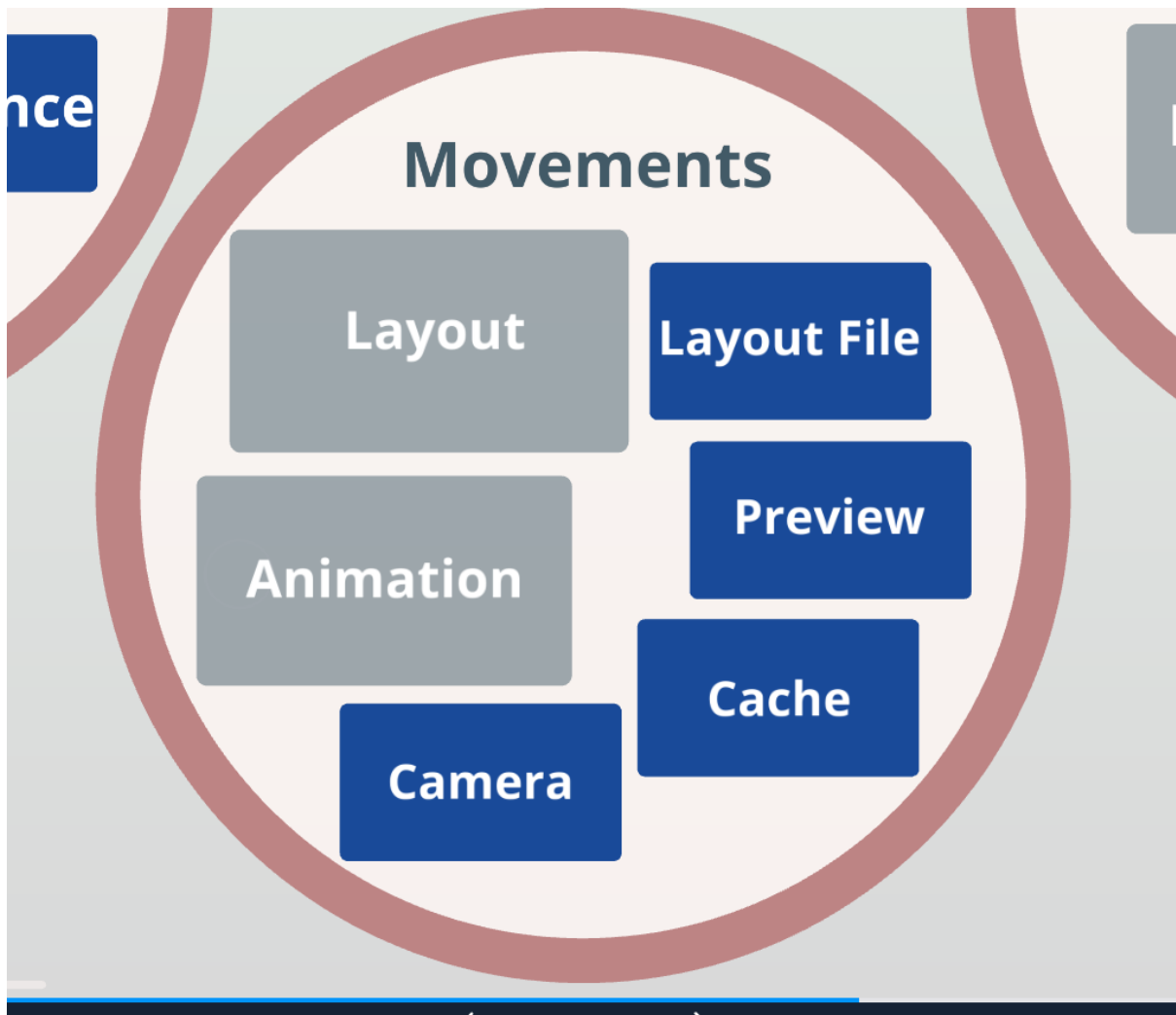


Fig. 11: The Layout and Animation Phase

Screen Shot

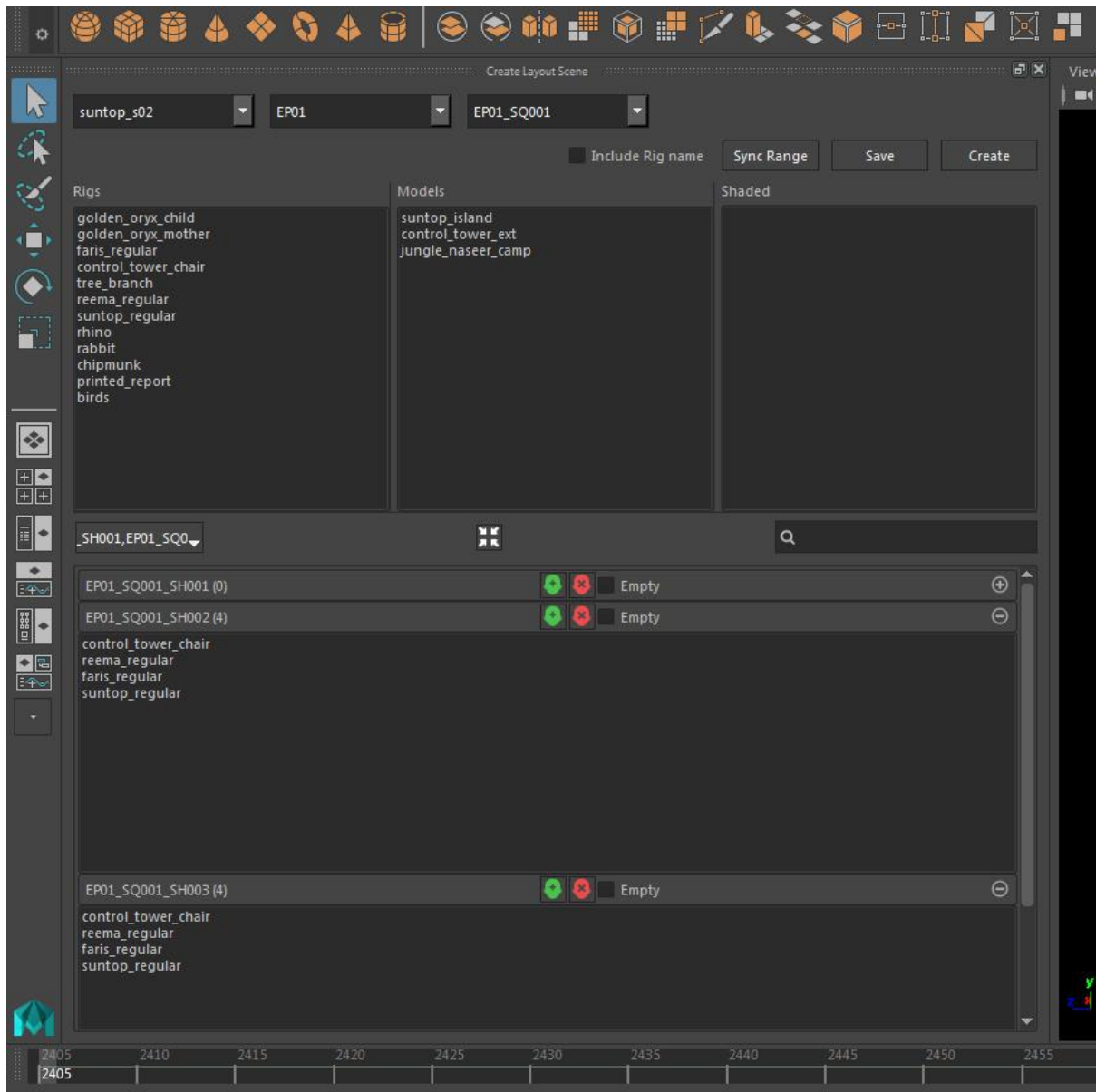


Fig. 12: The Create Layout GUI docked inside *Maya*

Features

The most prominent features of the *Create Layout* tool are as follows:

- Create the layout file for the sequence by *referencing* all the required assets from the published stream.
- Save the layout *Autodesk Maya* file to the sequence in the tactic server.
- Conform the frame ranges for the scene cameras to sync with those stored in the *Tactic* database.
- Export stills from each shot in layout as preview.

Hint: The *Create Layout* tool lists and references all the assets published to the episode of selected layout.

Usage

The typical workflow of the create layout tool can be as follows:

- Select the *project*, *episode* and *sequence* from the drop down list to list all the assets published to the selected *episode*.
- Select all the shots for which the layout needs to be created.
- Select and add all the required assets to each shot where they feature.
- Hit the *Create* button to import all the required assets including the environment.
- Place the camera and assets on the required location in the environment by moving and keying them in place. Pose the characters, if required.
- Hit the *Save* button and specify the suffix for the name of the layout.

Tip: Selecting all the shots from the drop down list is a standard practice when creating a single layout file for the whole sequence. It works generally because the assets throughout the sequence remain the same.

Tip: Often more than one layout files are created if the environment through the sequence changes. e.g. Interior and Exterior.

Requirements:

- *Autodesk Maya*
- *Southpaw Tactic*

4.3.2 Multishot Export

Multishot Export is a *Layout and Animation Tools* tool that runs inside of *Maya* that helps in exporting animation info from *sequence* level scenes or scenes that contain multiple shots. It works best when it is used on scene which have been created using the *Create Layout* tool.

Tip: *Multishot Export* works best when dealing with scenes created by the *Create Layout* tool

Screen Shot

Features

- List the shots present inside the current animation scene.
- Switch from one shot to another
- Export the following from all or multiple shots in your scene + *Point Cache* for each or selected *character*, *prop* and *vehicle* + *Preview* for each camera in the scene. + *Maya* and *Nuke* format camera animation.
- Burn overlays onto the preview showing useful information about the shot.
- Export the output directly to *tactic* server or export to a local directory.
- Defer the extraction to a *multishot Deadline* pool to save time for artists.

Tip: Save your machine time by offloading preview, cache and cam extraction tasks to deadline

Usage

- Fire-up the multishot export in an *animation* or *layout* scene, which has preferably been created using the *Create Layout* tool. Multishot will show the *project*, *episode* and *sequence* information stored in the scene.
- Select the shots you want to work with from the drop down list at the bottom.
- For each shot select the *rigs* whose cache you wish to export.
- For each shot select the *display layers* you want to make appear in the preview video.
- For each shot you can choose if you want to bake and export the camera.
- To perform extraction do:
 - Either Hit *Export* to begin extraction immediately
 - Or Hit *submit* to defer the extraction procedure to the *multishot deadline* tool.

Requirements

- *Autodesk Maya*
- *Southpaw Tactic*
- *FFMPEG*

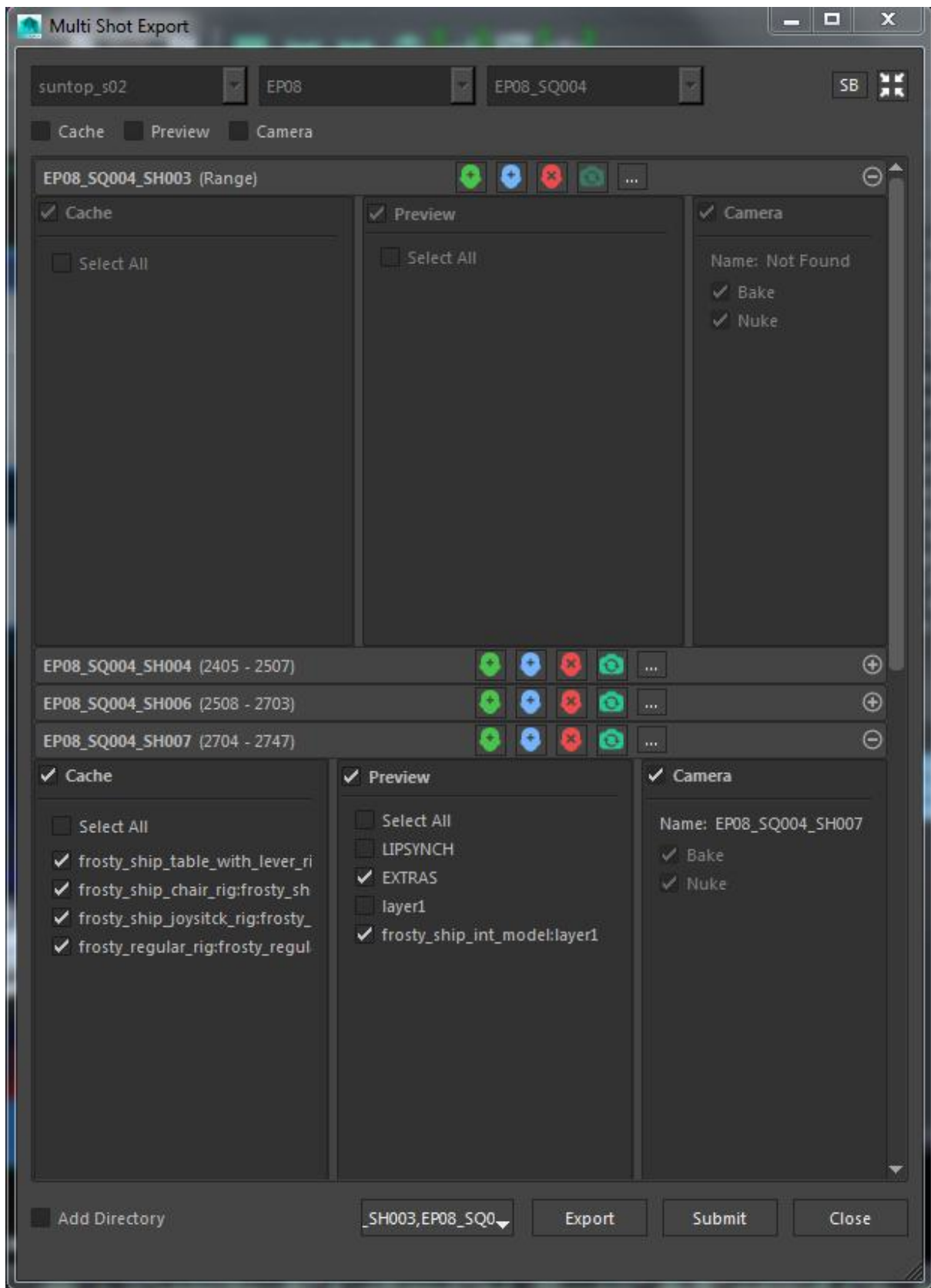


Fig. 13: Multishot Export tools running inside maya.



Fig. 14: Output of the multishot export showing info using *Overlays*

4.3.3 Mactor

Mactor or *MOCap TO Rig* is an *Animation* tools useful in *retargetting* motion capture or skeleton data from other sources onto a rig such as *Advanced Skeleton*

A popular use case is to use *Mixamo* downloads to map onto *advanced skeleton* <<https://www.animationstudios.com.au/advanced-skeleton>>. This functionality can be extended to other input and output hierarchies.

Screen Shot

Features

- Prepping Rig for retargetting
- Importing Motion Capture (skeleton-based) animation data into maya.
- Retargetting imported animation data on to a *rig* using Maya's *HumanIK* technology.
- Baking animation data on the rig controls.
- Clean up rig scene after animation import.

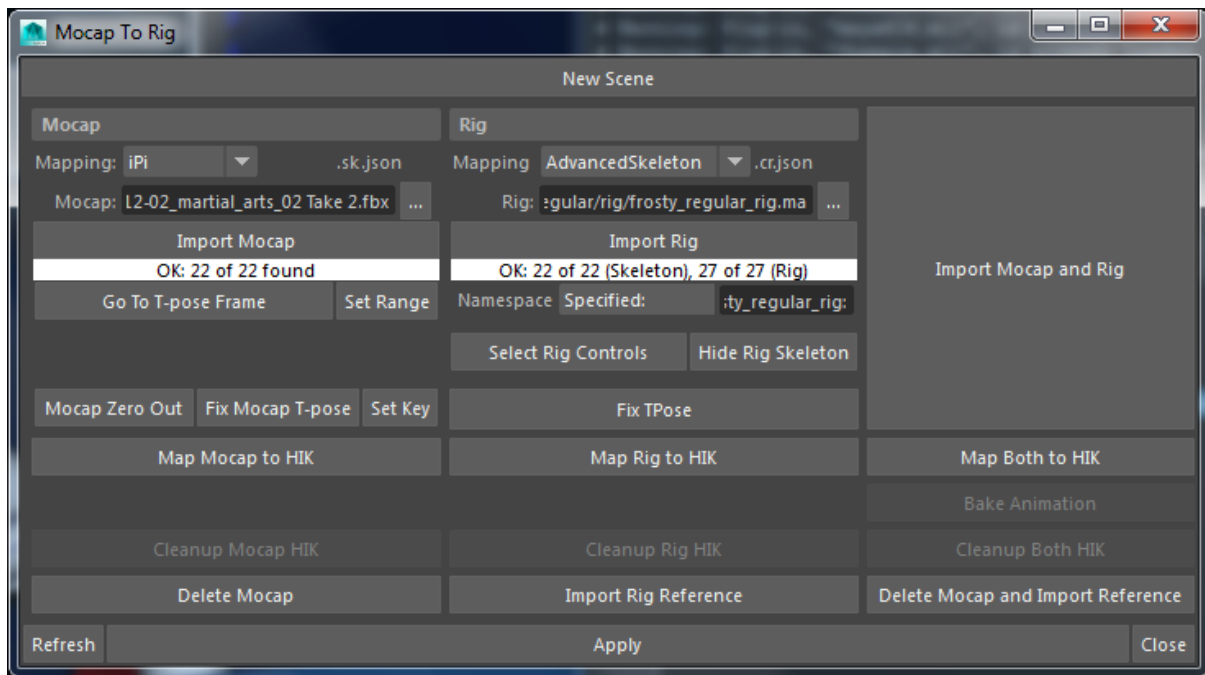


Fig. 15: Mactor (MEL-based) UI running inside of maya

Usage

- Specify the mocap mapping name from drop down menu
- Provide Path for the *fbx* file.
- hit `Import Mocap` to import the fbx file.
- hit `Goto T-pose Frame` to go to frame 0
- If the mocap data does not contain the T-pose frame on frame 0. You should create one by hitting `Mocap Zero Out`, `Fix Mocap T-pose`, `Set Key`
- Specify the *Rig* mapping.
- Provide the path for *rig*.
- hit `Import Rig` to *reference* the scene file onto the current scene.
- If the T-pose is not proper hit `Fix Tpose` button on the *rig* side to fix it.
- Now we have both mocap and Rig in T-pose. Now hit `Map Both to HIK` to perform retargetting onto the *rig*. The rig should now have animation mapped onto it.
- Hit `Bake Animation` and `Cleanup Both HIK` button to bake and cleanup the scene.
- Now you use `Delete Mocap` to remove mocap data from scene.
- If required use the `Import Rig Reference` button to import the rig and remove namespace.
- `Select Rig Controls` button can be useful when exporting animation data for posing tools such as [studio library](#)

Requirements

- *Autodesk Maya*

4.4 Lighting and Rendering Tools

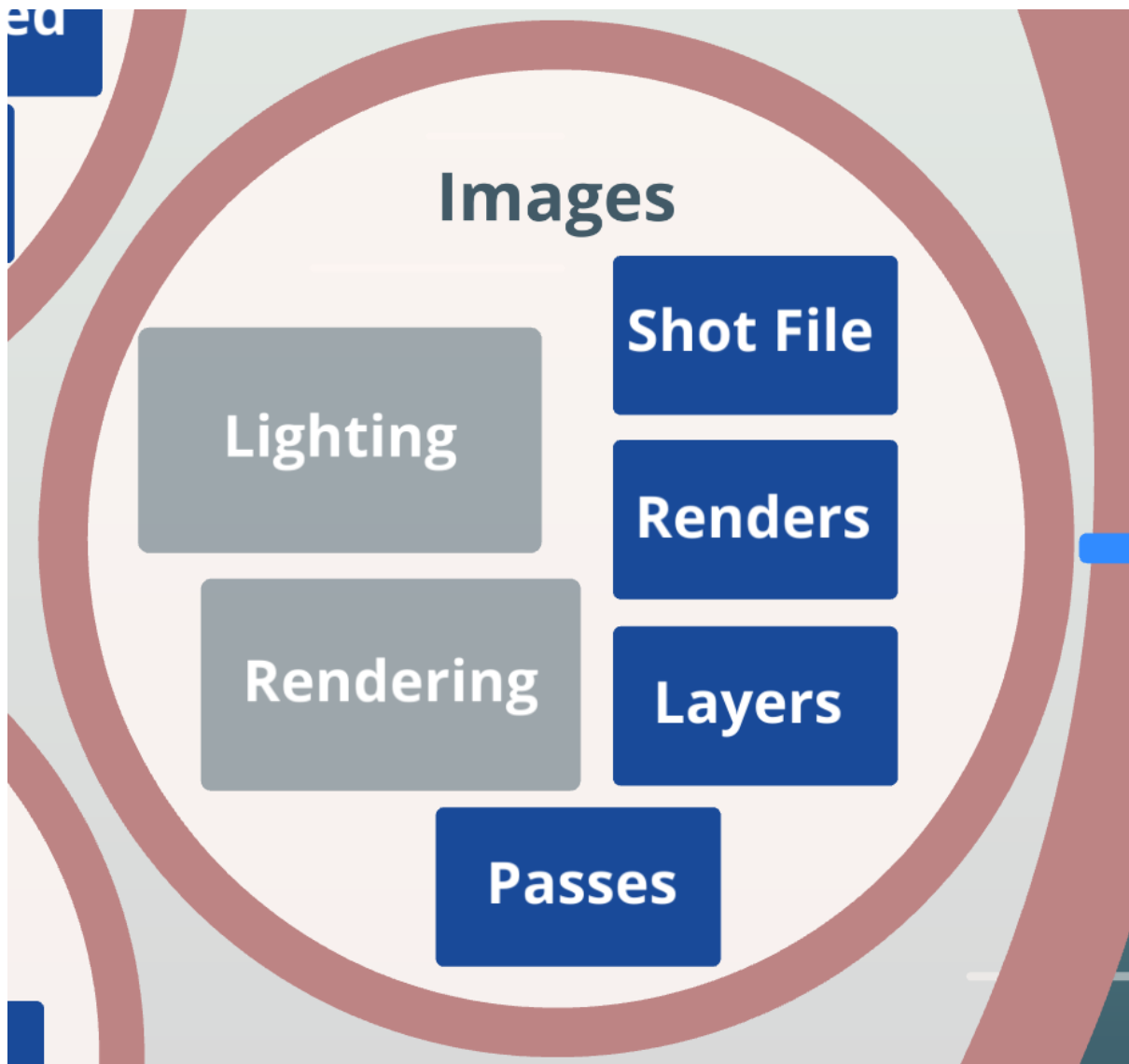


Fig. 16: The Lighting and Rendering Phases

4.4.1 Add Assets

Add Assets is a *Lighting and Rendering Tools* intended to be used at the time of lighting scene assembly. This tool runs inside of *Maya* in order to import assets into the current scene.

Screen Shot

Features

- List published assets for one or more sequences according to the specified context out of *rig*, *model*, *shaded* and *shaded/combined*
- Select and Reference all the required assets with multiplicity

Usage

The add assets tool fetches the projects and sequence from the and when the user selects a sequence to work with, all the assets available for that sequence are listed.

- Select a project from the drop down, the episodes drop down will be populated from the *Tactic* server.
- Select an Episode from the drop down to populate the sequences.
- Select one or more sequences to list all assets of the given context.
- Change the context from the drop down on the bottom to fetch the list of relevant files.
- Select the required assets, specifying the number of instances in front of it.
- Hit the *reference* button to reference all the specified assets into the scene in one go.

Requirements

- *Autodesk Maya*
- *Southpaw Tactic*

4.4.2 Setup Master Scene

Setup Master Scene is an *Production Planning* tool that runs as a standalone application that helps to associate assets to sequences. This is a necessary step to perform before creating layouts using the *Create Layout* tool.

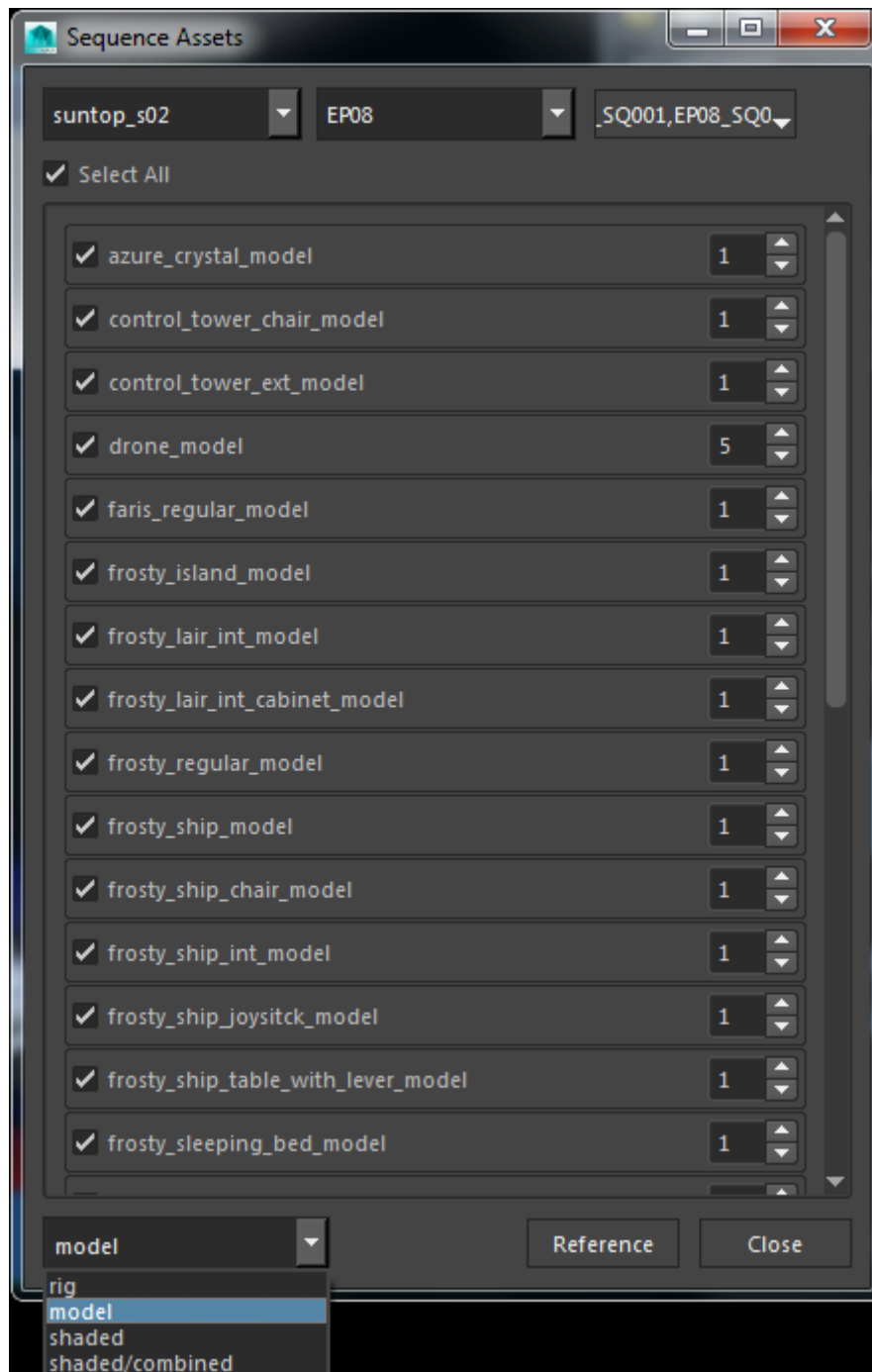


Fig. 17: Add Assets UI running inside of maya

Screen Shot

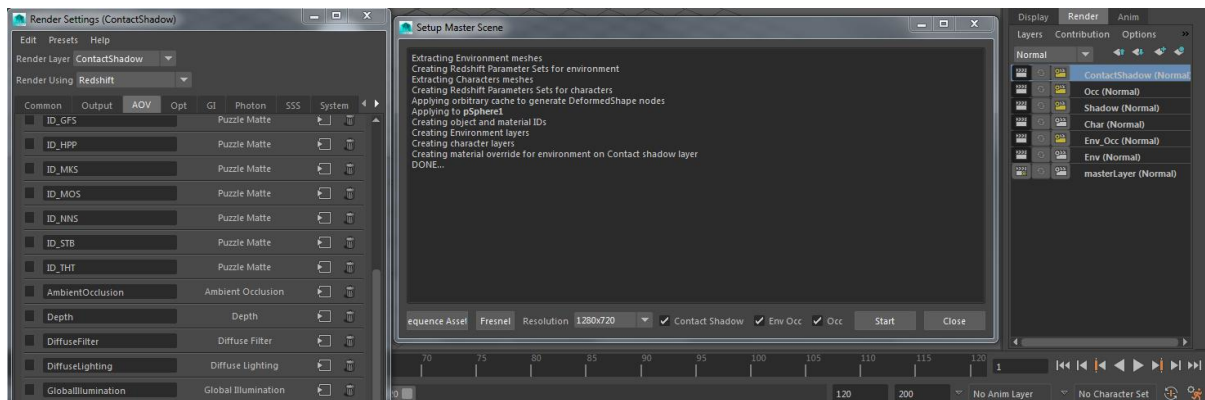


Fig. 18: Setup Master Scene UI from Inside of *Maya*

Features

- Prep the Environment in the scene for Rendering
 - Setup *Beauty* and *Occlusion* render layers with proper material overrides.
 - Add relevant *lights* and *meshes* to the scene
 - Setup *parameter sets* for the environment such as *Env_Smooth_Set*, *Env_Matte_Set* and *Env_Vis_Set*
- Setup Characters for Rendering
 - Setup *Beauty* and *Shadow* render layers.
 - Add character *Lights* to the render layers
 - Add passes such as *Puzzle Matte* object and material Ids.
 - Setup environment parameter sets for interaction with environments
 - Setup *parameter sets* for the characters such as *Char_Smooth_Set*, *Char_Matte_Set* and *Char_Vis_Set*

Usage

1. Add all the required assets to a maya scene using the *Add Assets* tool which can also be invoked from the *Sequence Assets* button.
2. Create a group called `characters` and add all the mesh node of the *characters*, *props* and *vehicles* to this group
3. Create a group called `environment` and add all the mesh node from the environment to this group.
4. Create a group called `env_lights` and add all the environment lights.

5. Create a group called `char_lights` and add all the character lights.
6. Choose the resolution from the drop down list and Check all the options you want in the setup!
7. Hit the `Start` button to setup the master scene.
8. If so required, add an optional *fresnel* render layer by hitting `Fresnel` button.

Requirements

- *Southpaw Tactic*
- *Autodesk Maya*

4.4.3 Create Shots

Create Shots is a *Lighting* tool intended to help lighting artists in creating the scene file for each *shot* after a master file for the *sequence* has been created.

Screen Shot:

Features:

- Generate multiple lighting scene assembly for shots from a master shot.
- Automatically find and replace available caches.
- Create collage using the output of stills from the shots in order to check the files.
- Modify mapping between objects and caches found on storage.

Usage:

1. Create a *sequence* “Master Scene” using *Add Assets* and *Setup Master Scene* scripts.
2. Provide a shots dir to scan and populate shots in the drop down.
3. Select the shot(s) for which the scenes are to generated.
4. Check on *Create Files* and *Create Collage* as required.
5. Hit *Start*.
6. When the mapping UI shows up adjust the mappings and select the appropriate name for the layers from the list. Press `OK` to proceed.

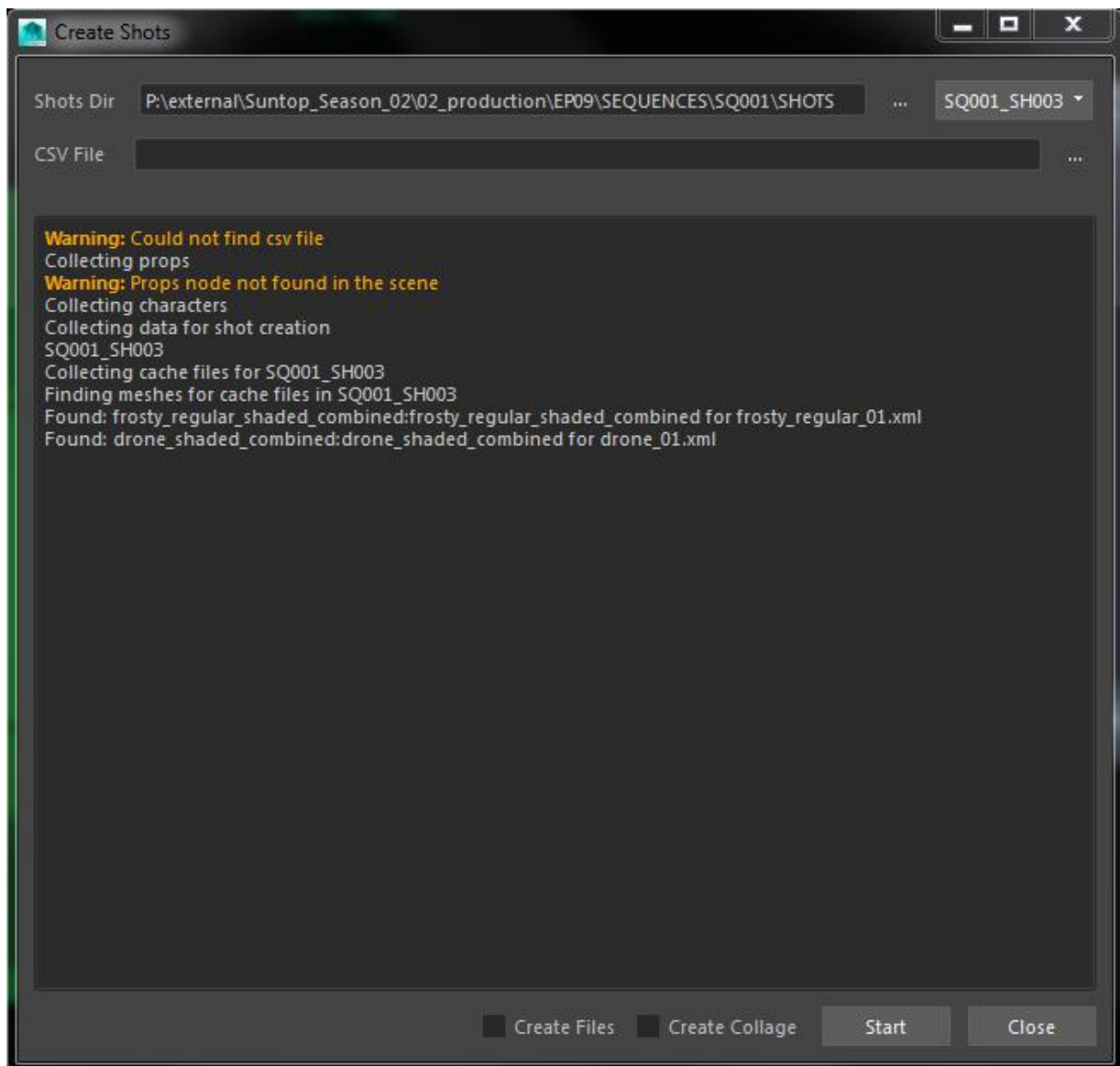


Fig. 19: The Create Shots tools running inside *Maya*

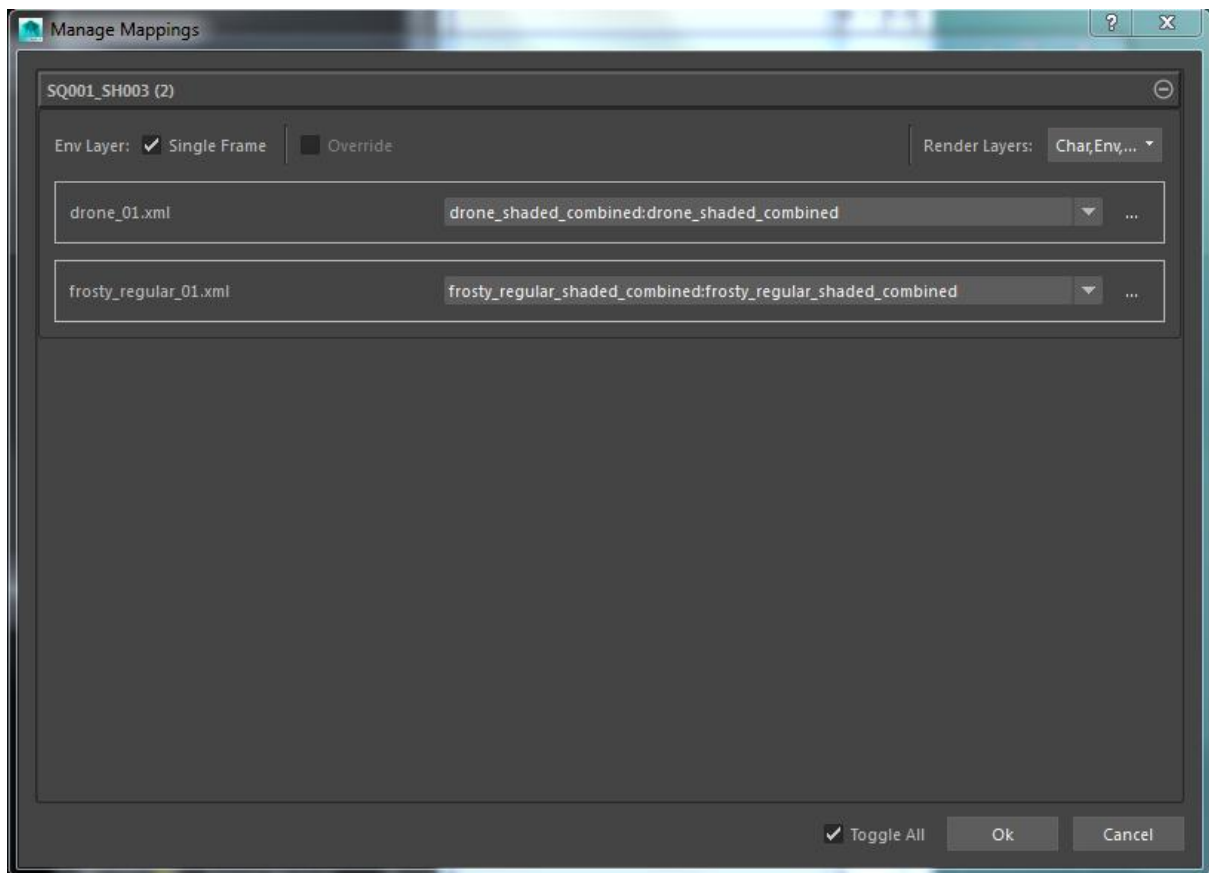


Fig. 20: UI for modifying the mapping for create shots

Requirements:

- *Autodesk Maya*
- *Window OS*
- *Image Magick*

4.4.4 Matte IDs

Matte IDs is a tool for *Lighting and Rendering Tools* artists that runs in *Maya*. It is useful in helping organizing object and material ids into AOV channels.

Screen Shot

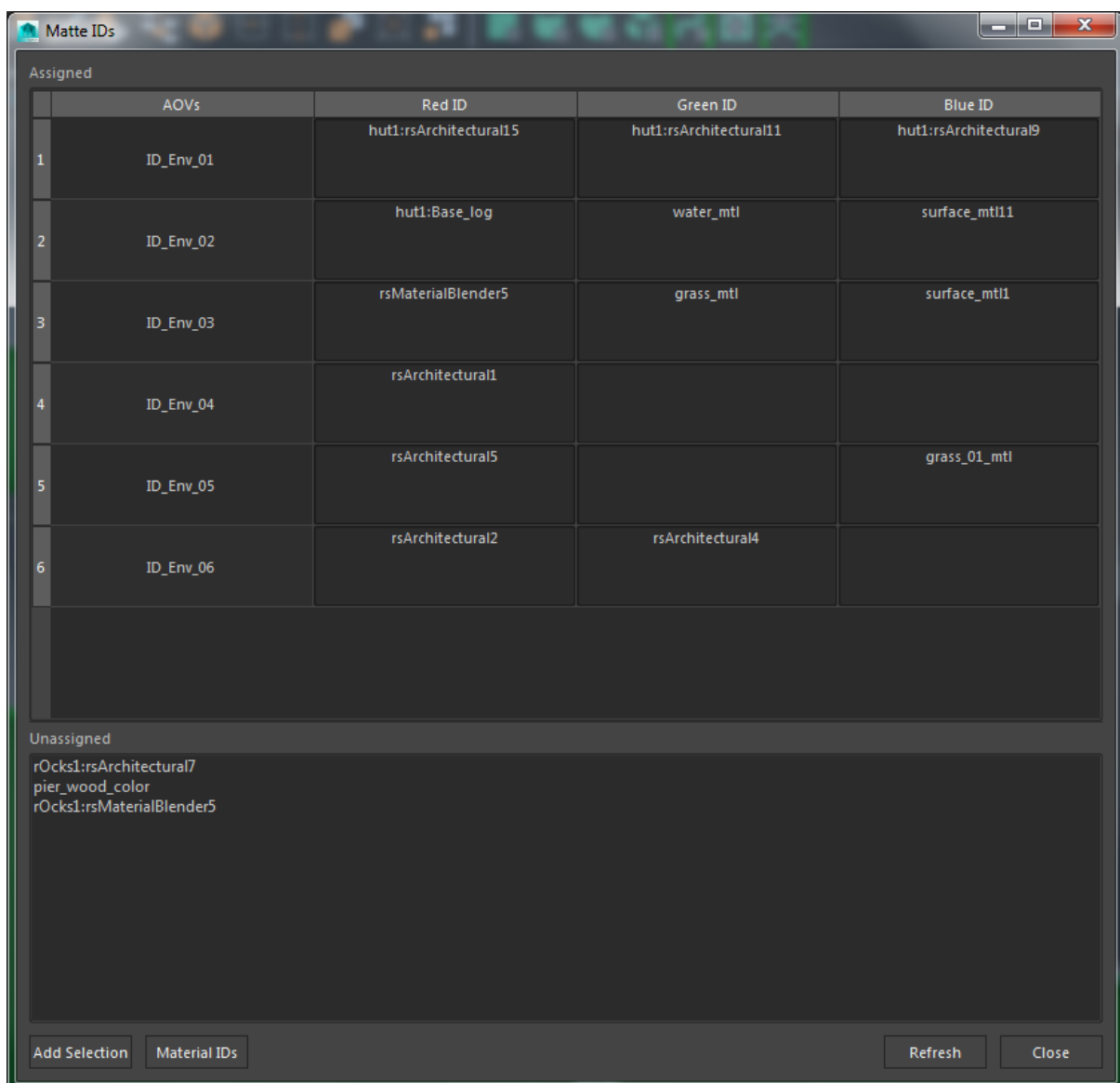


Fig. 21: The Matte IDs UI running inside of *Maya*

Features

- Create New *Matte AOVs* of *material* or *object* type.
- List Unassigned materials and objects.
- Assign Material to each channel of the id.

Usage

- Switch Mode to *Material IDs* or *Object IDs* by hitting the button at the bottom.
- Hit *Refresh* to List all the assigned and unassigned materials / objects
- Select Unassigned materials from the list on the bottom and hit *Add Selection* to assign redshift AOV channels to the ids
- *Drag and Drop* in the table to shift around IDs in AOVs.

Requirements

- *Redshift 3D*
- *Autodesk Maya*
- *Southpaw Tactic*

4.4.5 Redshift AOV Tools

Redshift AOV Tools is a *Lighting and Rendering Tools* collection of some simple but useful **important** automations which help you in building *redshift* rendering scenes. This tool also helps in setting up rendering standards that are of great use in *Post Production*

Screen Shot

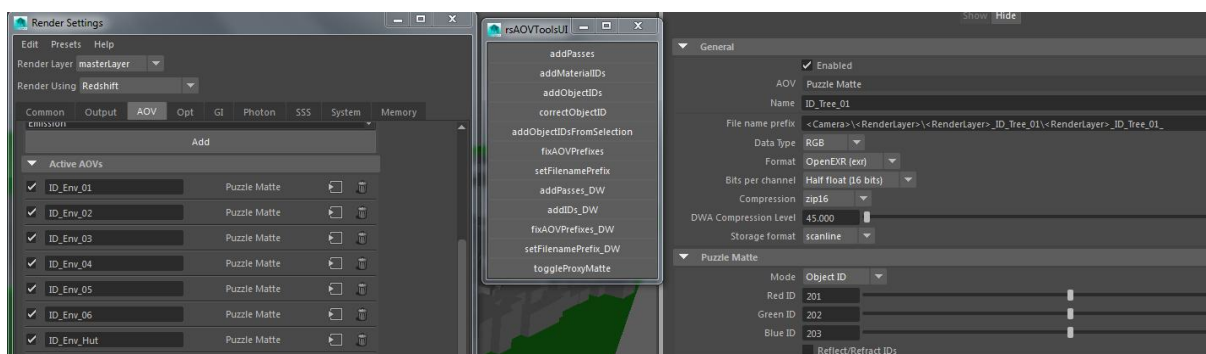


Fig. 22: Redshift AOV Toolbar in the middle

Features

- Add all the required passes to the scene as required
- Add Matte ID AOVs such as *Material IDs* and *Object IDs*.
- Set the AOV and filename prefixes in order to have consistency in render output naming.

Usage

- Click on the buttons to perform the changes in a *maya* scene.

Requirements

- *Autodesk Maya*

4.4.6 Pre-CC

Pre-CC is a tool used by rendering artists to automatically composite the render output to allow to check visually for errors and missing frames.

Note: This program invokes *Nuke* in the background to perform the composite.

Screen Shot

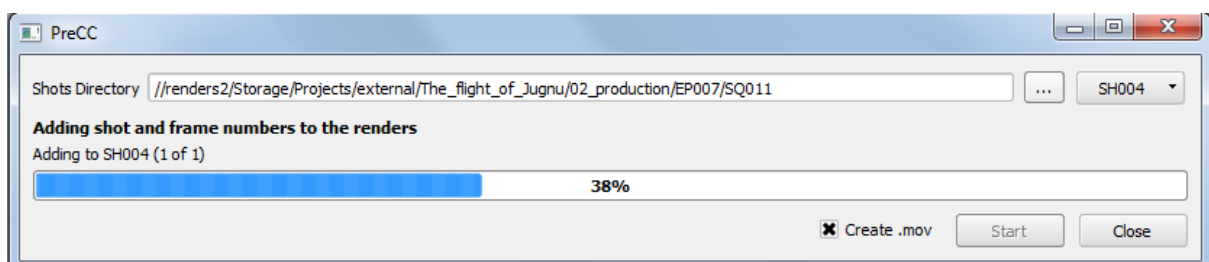


Fig. 23: Pre-CC Screen Shot

Features:

- Generate composited outputs from renders from sequences by scanning the file system.
- Add overlays on the output describing the *shot* and *frame #*
- Ability to control composite method using underlying nuke scene



Fig. 24: Output of *Pre CC* played in QuickTime with overlays

Usage

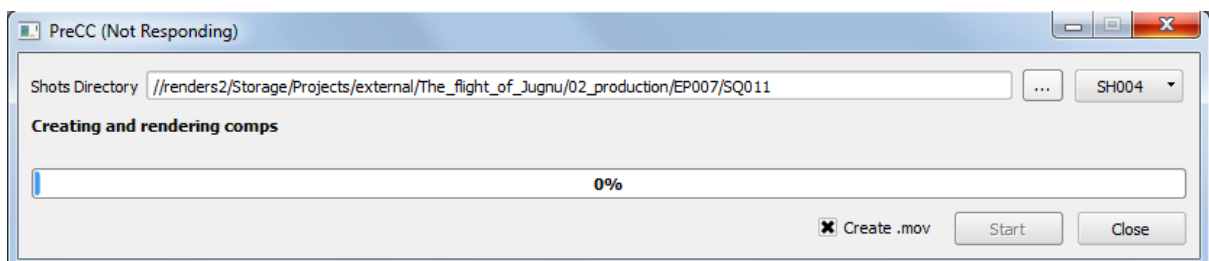


Fig. 25: Pre-CC while creating comps

- Provide the path for a sequence for render location.
- Select shots that need to be checked
- hit *Start*.

Requirements

- *Windows*
- *Nuke*

4.4.7 Scene Bundle

Scene Bundle is an *rendering* tool that runs inside of *Maya* and as well as a standalone application that helps rendering artists to bundle their scenes by copying all its requirement into a single packet and sending them as source for rendering jobs on the *deadline* cluster.

Screen Shot

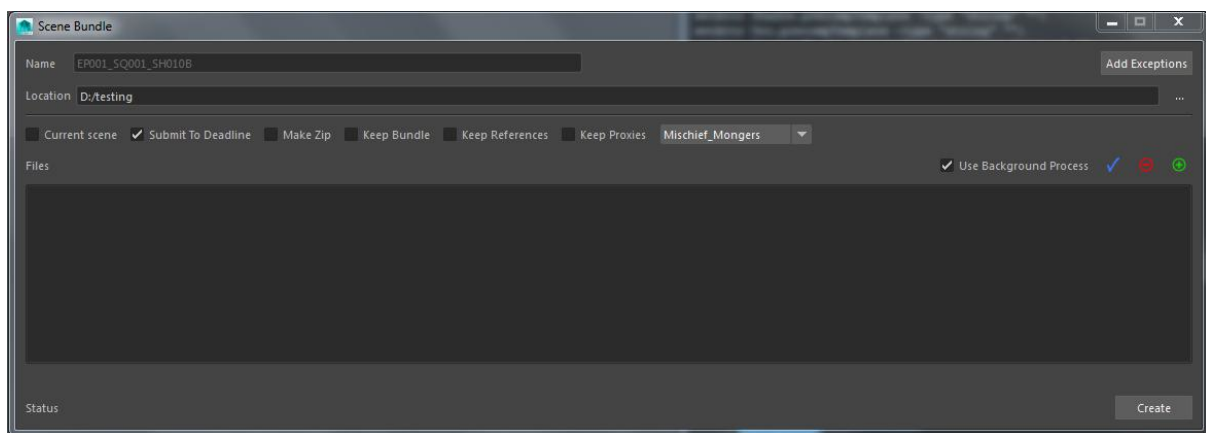


Fig. 26: Remap Textures running inside maya.

Features

- Collect all dependencies from a rendering scene to a single *maya* project folder called a *bundle*.
- Create an archive from the *bundle*
- Copy the *bundle* to an appropriate location ready for rendering.
- Send a job to the *deadline* according to predefined parameters.

Usage

1. Provide a name for the bundle.
2. Provide a location for the bundle most suitable for collection. This location should have ample disk space to collect all the dependencies.
3. If you intend to bundle the current scene check `current scene` check else provide a list of scenes to process.
4. If submitting to deadline provide the *project*, *episode*, *sequence*, and *shot* information either by drop down or by clicking on the list and using the UI that appears.
5. Hit `Create` to start the bundling process.
6. Attend to ignore or raise the appearing error prompts if any.

Requirements

- *Autodesk Maya*
- *Thinkbox Deadline*

4.5 Post Production

4.5.1 Backdrop Tool

The *Backdrop Tool* is a set of compositing scripts that help automate the process of compositing from inside *Nuke*.

Screen Shot

Features

The features have been presented below under *Backdrop Tools* and *Read Node Tools* according to the context of their actions.

Tip: Arrange all the nodes from one shot into a single `Backdrop` node in order to gain advantage of all the *Backdrop Tools*

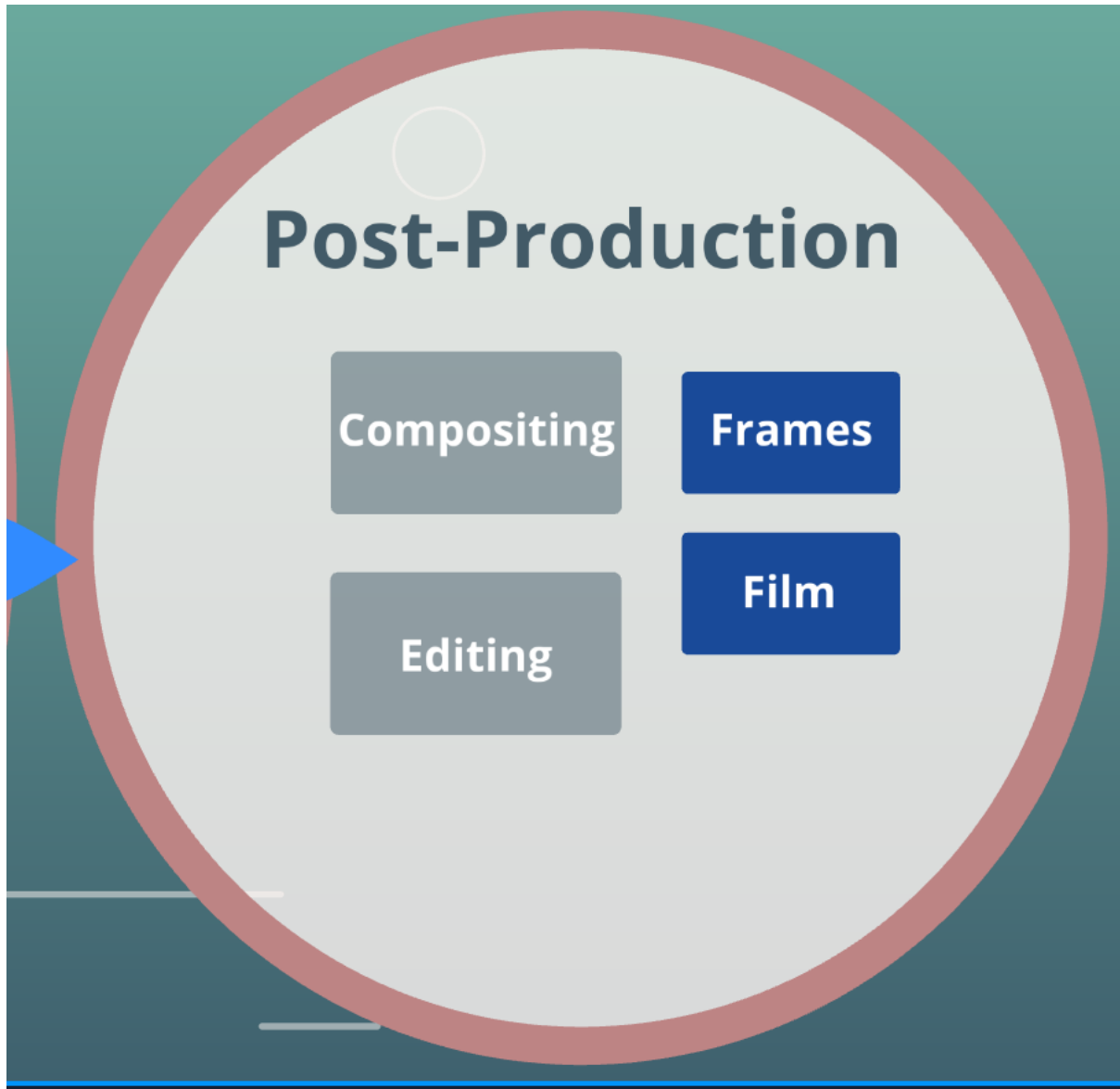
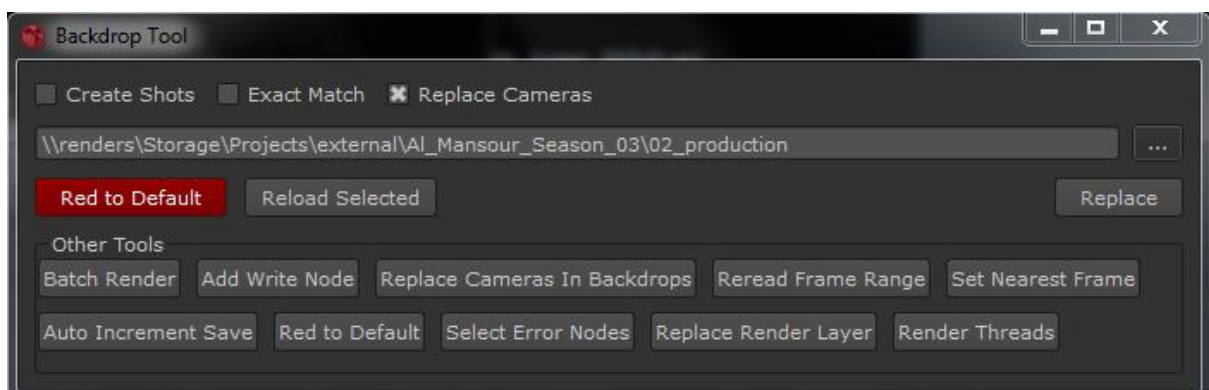


Fig. 27: The Post Production Phase

Fig. 28: The Backdrop tool UI inside of *Nuke*

Backdrop Tools

Replace Read Path Given a backdrop containing the network of nodes that can render one shot perfectly, The *Replace Read Path* feature can either replace all the relevant knobs of the read and write nodes with the render output related to another shot or replicate this layout for other shots by scanning the render output on the provided path.

Add Write Node Given a node under selection one can add a write node after it and replace its knobs with the appropriate ranges and output paths.

Replace Cameras in Backdrop Given a backdrop containing a camera node, the node is replaced by appropriate camera for the given shot.

Replace Render Layer Detects all the render layers being used in the current selection / backdrop and replace the render layers with the other ones that are available in the file system containing the render outputs

Read Node Tools

Select Error Nodes Select all the Read nodes which have error.

Tip: Use the *Select Error Nodes* feature to select all the nodes that have error and then use them in tandem with the rest of the tools to change them all at once.

ReRead Frame Range Update frame range on the selected read nodes by scanning the filesystem.

Set Nearest Frame Change the on error behaviour of the selected nodes to nearest frame.

Auto Increment Save Automatically save the current nuke script after the given interval.

Red To Default Change the color of the *red* or error nodes to default.

Hint: All the tools present under *Read Node Tools* work selected read nodes in nuke.

Third Party Tools

The following 3rd Party Tools can also be invoked from this UI for easy access.

RenderThreads This is a popular rendering scripts available on [Nukepedia](#) which can be used to perform renders asynchronously

Requirements

- *The Foundry Nuke*

THE LAST WORD

This guide is just an introductory documentation of the tools at hand. There is need for more literature in terms of more user and developer documentation which is required to make a complete product out of this effort.

As more well rounded efforts are done to standardize the underlying library and more back-ends are integrated, the need for comprehensive documentation is also mandatory.

Please send any questions and corrections to [Talha Ahmed](#)