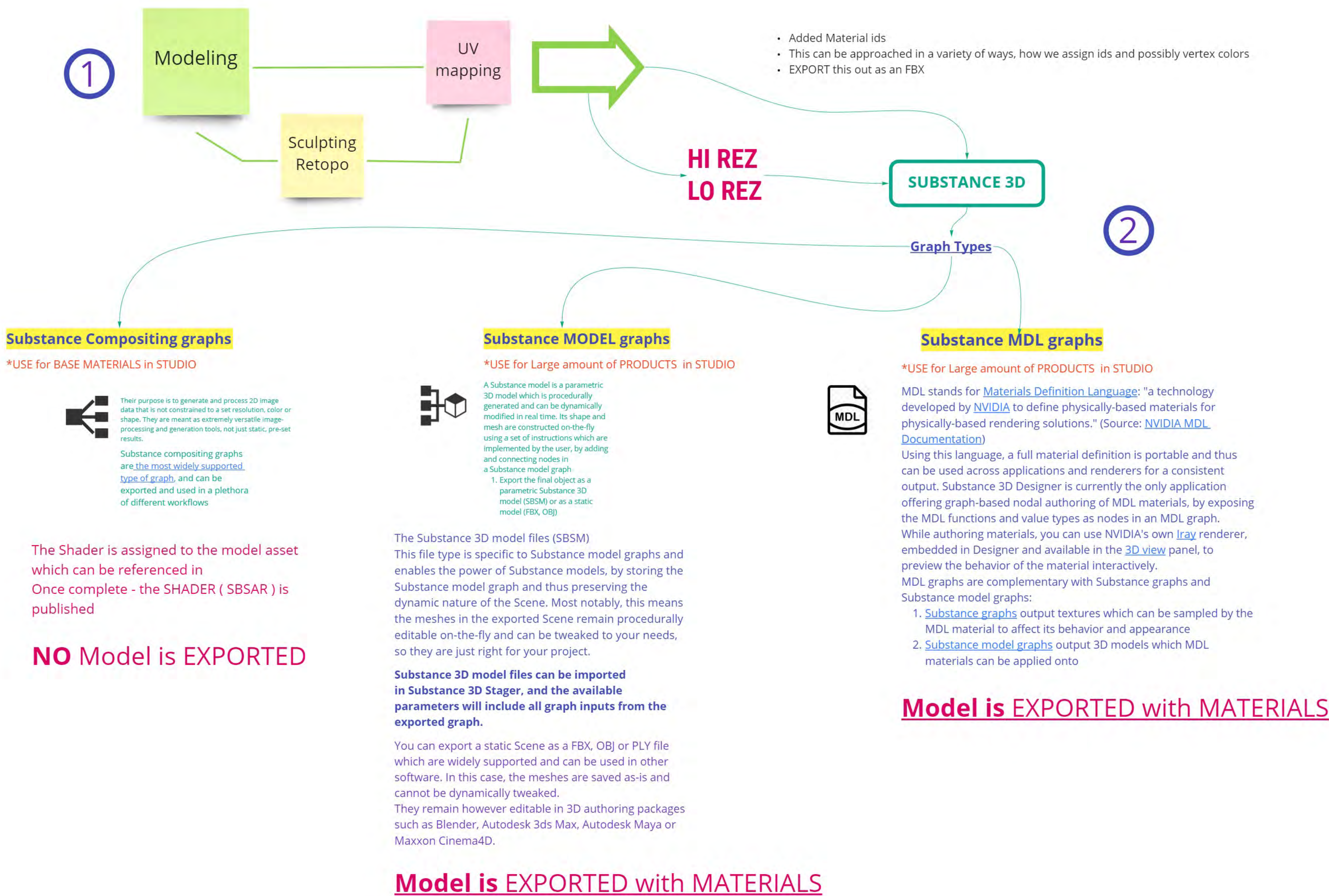




Designer Pipeline setup

3

Identify and group what settings to add into your Project files. This is different for every studio and requires a certain amount of planning! In almost every case at least 2 Projects should be defined: one for global, studio-wide defaults (like standard templates, shader files, baking settings), and one with more specific content such as Library content. If you have different projects running simultaneously you might want to create multiple project configurations for each (so 3 or more total).

The difference between exporting Substance 3D model files and static 3D files is akin to the difference between publishing Substance 3D asset files and exporting bitmaps from Substance graphs. One outputs dynamic files which can greatly accelerate workflows, while the other outputs static files which are universally supported.

A Large part of this so far can be automated as well

This is different for every studio and requires a certain amount of planning! [Using Project Configuration Files - SBSPRJ](#)

Create an [SBSCFG configuration](#) file listing all project files, place it under version control. If you have multiple projects, you can create a Config for every project. Listing all project files, place it under version control.

Designer Pipeline setup
Every type of file is explained with more detail on child pages of this page, but the short overview of how to ideally define a custom setup for Designer is as follows:

1. Identify and group what settings to add into your Project files. This is different for every studio and requires a certain amount of planning!
2. In almost every case at least 2 Projects should be defined: one for global, studio-wide defaults (like standard templates, shader files, baking settings), and one with more specific content such as Library content. If you have different projects running simultaneously you might want to create multiple project configurations for each (so 3 or more total).
3. Create the relevant [SBSPRJ files](#) and place them and their content under version control. It is strongly recommended to separate Designer pipeline and library content from your actual project content and resources (3D models, textures, code) by creating a separate repository for it.
4. Create an [SBSCFG configuration](#) file listing all project files, place it under version control. If you have multiple projects, you can create a Config for every project.
5. Set up every user's [User Preferences.xml](#) to reference their relevant configuration file.
6. You can have every user do this manually, or you can script this by injecting lines into their XML file. [More info on the relevant page.](#)

The Pipeline PRESETS can do tasks like baking and updating bakes per asset

Scons

Python based build system

◦ Allows implementing operations and control the build using python

Sophisticated dependency control

This is a sample

◦ There are tons of build processes in a game studio

◦ Try to use what is there rather than building your own



4

Within SUBSTANCE - The materials

Materials
for Client



Substance Painter



- Procedural
- From Substance Library

Substance Designer



- Procedural
- Scanned Material
- From Substance Library
- Hybrid - Scanned and procedural
- Photogrammetry

SUBSTANCE SAMPLER - Single Source

The features in Substance 3D Sampler give you a complete material digitizer toolbox.

AI-powered scan processing

The Image to Material feature removes shadows and generates albedo, roughness, normal, and displacement maps with unparalleled accuracy.

Open ecosystem

Use your content anywhere. Send your assets directly to Painter and Stager, or use them in your favorite 3D applications.

IBL creation

Create HDR environment lights in real time from 360° images. Enrich and edit them with parametric effects or even other lights.

Edit in Photoshop

Send any input image to Photoshop and take advantage of its extensive image editing toolset, with direct feedback in Sampler.

Atlas Splatter

Scatter leaves, pebbles, and more on your material. Choose from a vast library of atlases made by our team of 3D experts or use your own scanned content.

Parametric effects

Add filters. Combine weathering and natural effects such as moss, snow, water, or dirt with any existing material.

5

SUBSTANCE REVIEWING - The materials



CAN MAKE updated revisions that
down the pipeline CAN
automatically review

FAST ITERATIONS

Export out as SBSAR to MAYA - 3D apps

- 3D Painting
- Live Link to Substance Designer
- BAKE Detail Normals
- Displacement
- Curvature
- Ambient Occlusion
- World Space Normals
- Id Maps

Export out as SBSAR to MAYA - 3D apps

- Dynamic Textures
- Dynamic Resizing
- Procedural Material Workflow
- BAKE Detail Normals
- Displacement
- Curvature
- Ambient Occlusion
- World Space Normals
- Id Maps

Export out SBSAR / model / Texture MAPS

Export out SBS / SBSAR / model / Texture MAPS

Export out To STAGER



REVIEW ASSETS

EXPORT OUT GLB'S

Substance 3D Stager a professional staging tool for scene design and rendering.

Import content, arrange your scene, apply materials and textures, adjust both image-based and physical lighting, save cameras with different resolutions, and render photorealistic imagery - all in Stager!

Lighting made easy

Enhance your lighting with the environment light editor or add physical 3D lights to your scene for advanced lighting design.

Diverse format support

Enjoy support for an array of formats from CAD (for enterprise customers) to USD and glTF, as well as parametric Substance materials, lights, and models.

Physics awareness

Set up realistic collisions between objects during positioning and transformation.

Project showcase

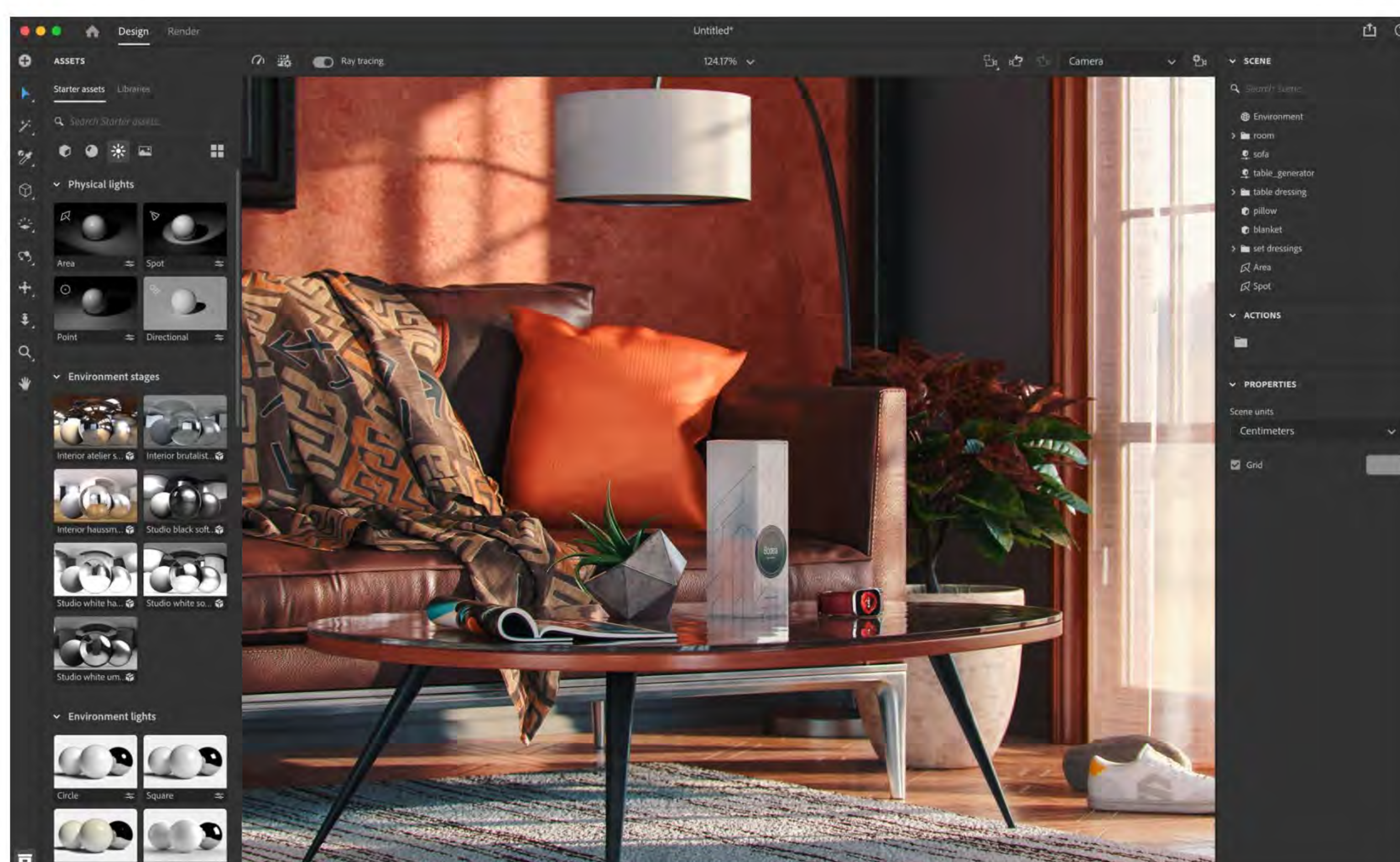
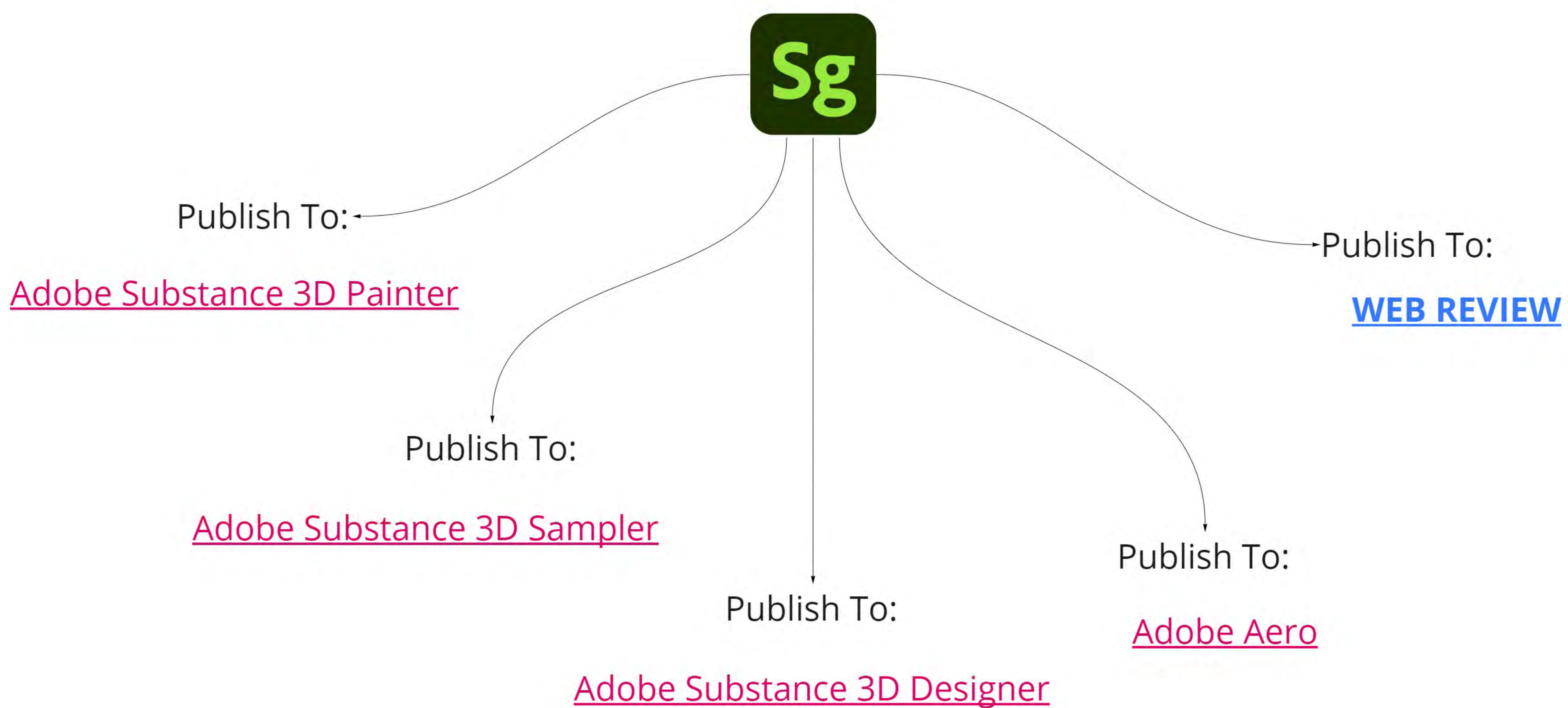
Import your Painter projects into Stager with a single click and pick up where you left off.

Interactive workspace

Switch between real-time rendering and interactive path tracing while keeping a high visual quality.

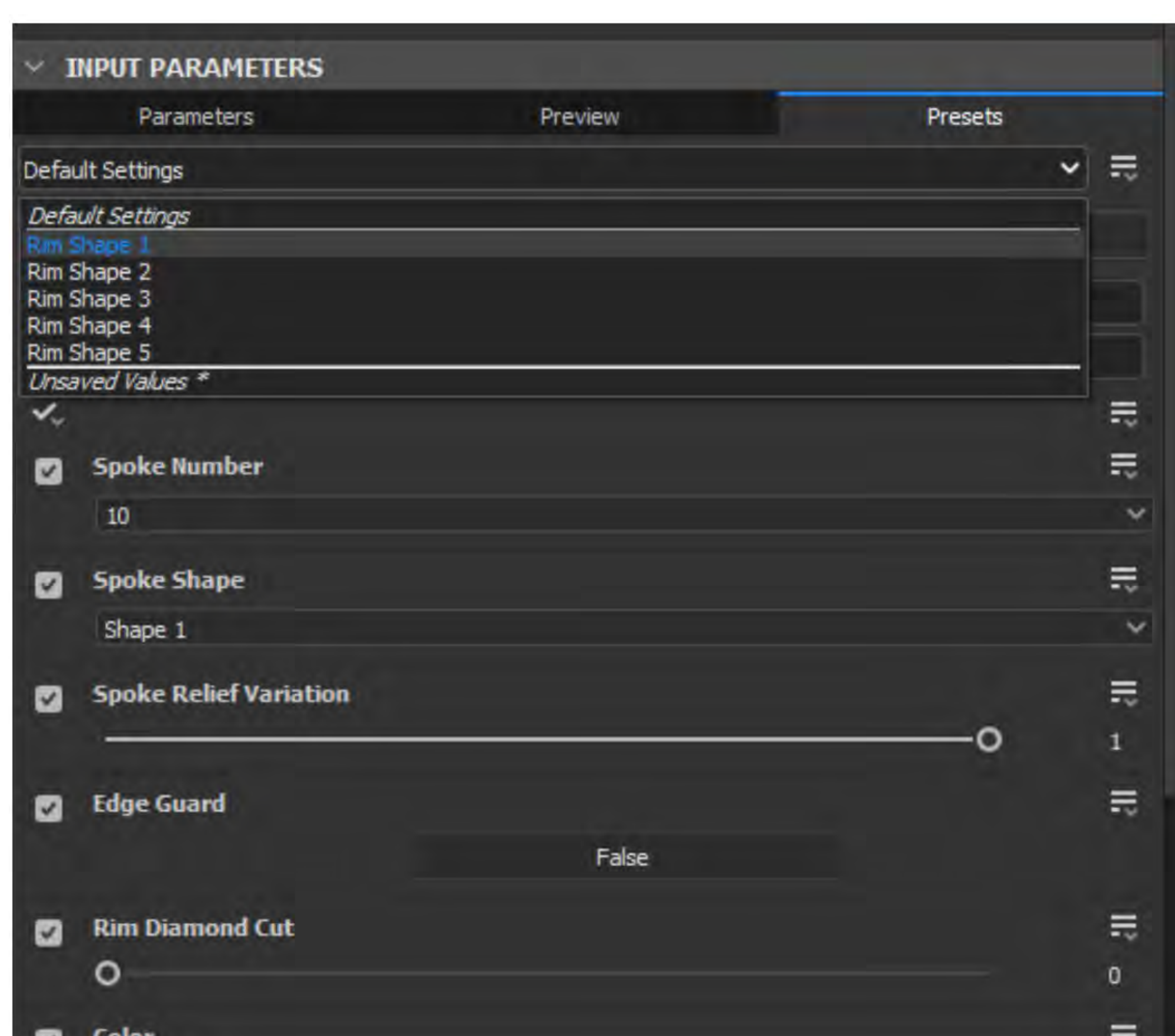
CLIENT and INTERNAL Approval

Export out online - published viewer - and save to library -

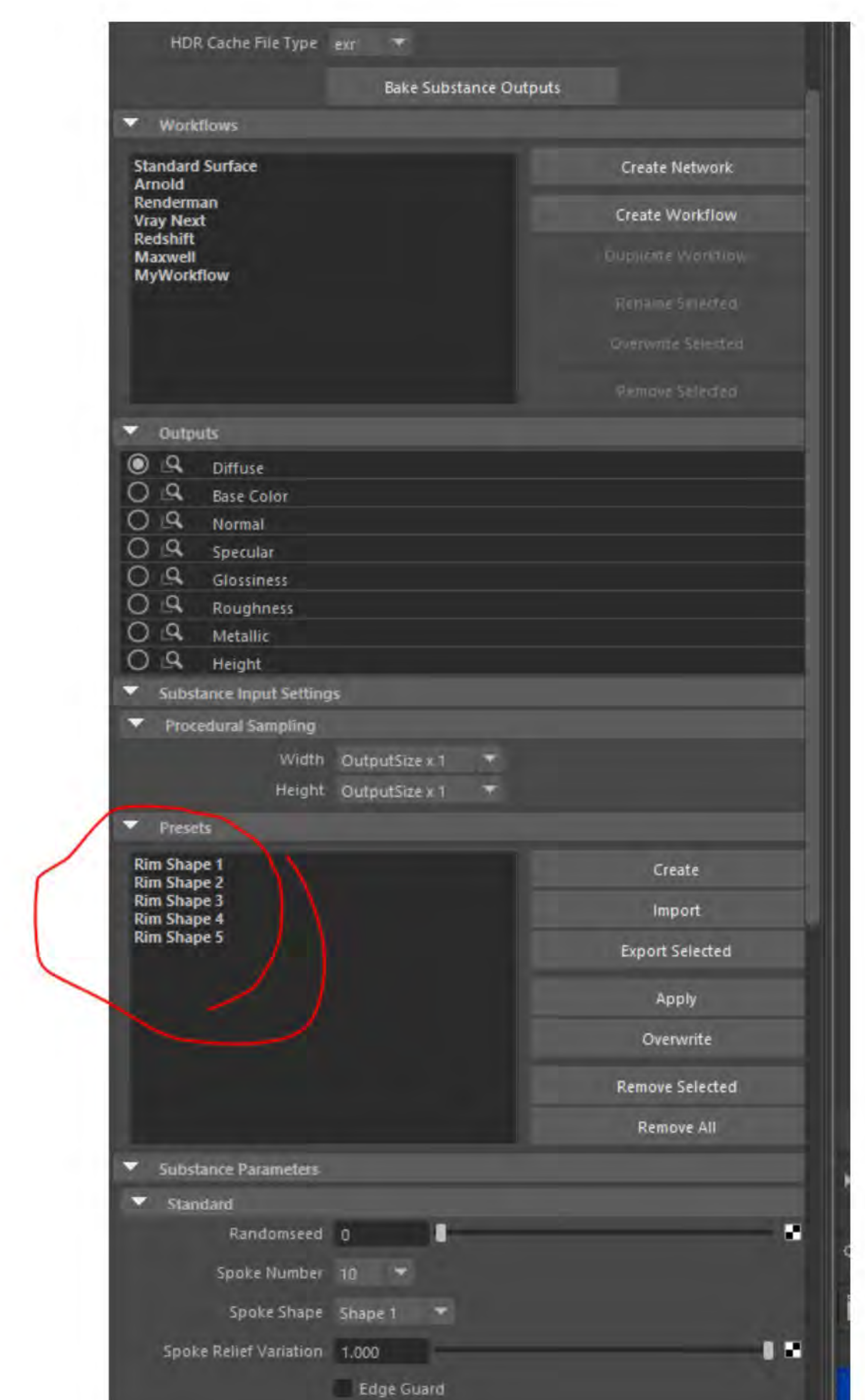


Within The Substance World There are many ways to get a material.

- 1- Make it in PAINTER and publish maps - Compile in Designer as a SBSAR - EXPORT out
- 2- Make it in DESIGNER and publish as a SBSAR - EXPORT out
- 3- Bring a Premade Shader from Substance Source into DESIGNER
- publish as a SBSAR - EXPORT out
- 4- COMPILE MULTIPLE SHADERS in DESIGNER and publish all of them as ONE SBSAR - Then export EXPORT out
- 5- COMPILE ONE TYPE of SHADER in DESIGNER, from Various Sources all with Presets in them with in Substance Designer. Publish all of them as ONE SBSAR
- Then export EXPORT out.



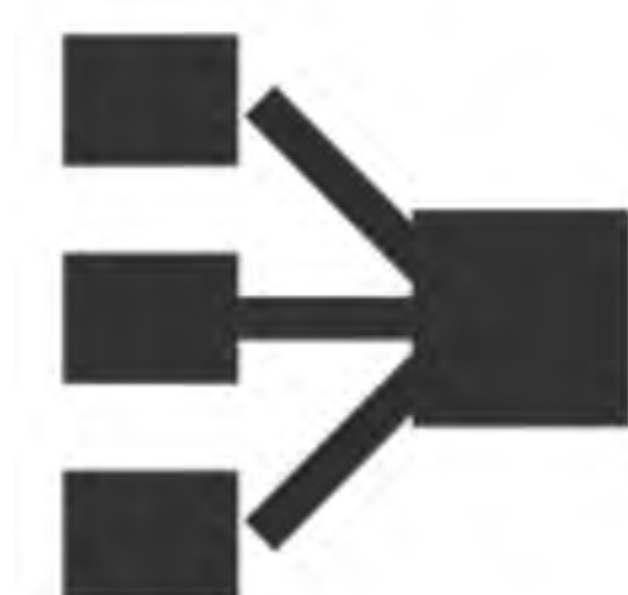
** 05 can best be seen in the Substance Source Material - with multiple options and pull down features made for one single Car Rim. All of these Transfer over to Maya for Example



PRACTICAL USES

01 - Compile All - Show Specific

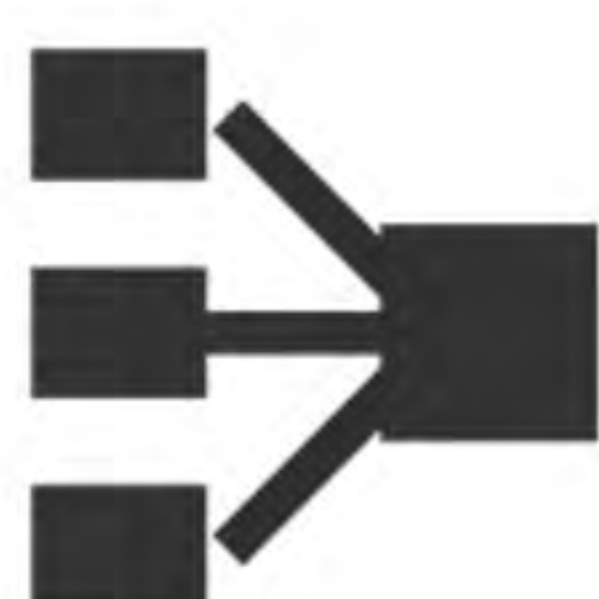
SBS - Shader 01
SBS - Shader 02
SBS - Shader 03
**PROCEDURAL
MAPS**



-----**.SBSAR**-----> MAYA, UNREAL etc

02 -Solo Individual Shader (with sub presets)

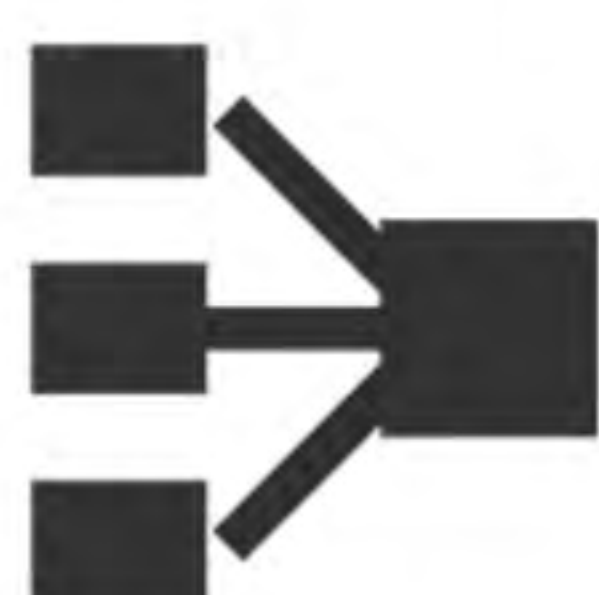
PROCEDURAL
SBS - Shader 01



-----**.SBSAR**-----> MAYA, UNREAL etc

03 -Solo Individual Shader (with sub presets)

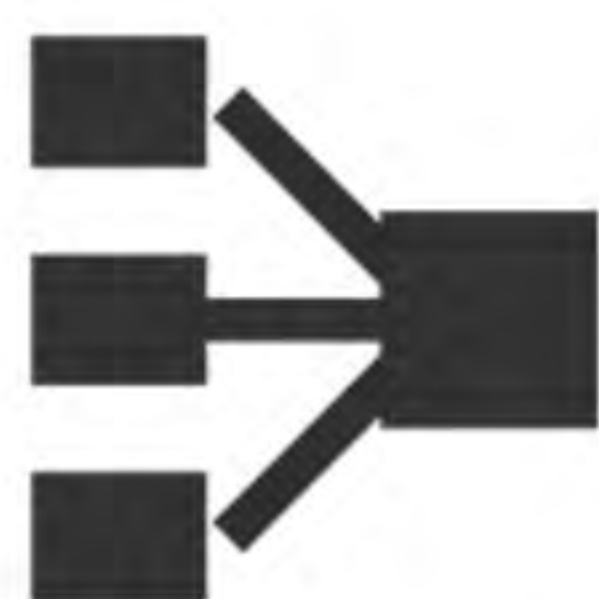
MAPS
SBS - Shader 01



-----**.SBSAR**-----> MAYA, UNREAL etc

04 -Solo Individual Shader (with sub presets)

MAPS
SBS - Shader 01



-----**.SBSAR**-----> MAYA, UNREAL etc

05 -3D Models and Materials

MAPS
SBS - Shader 01



-----**.SBSAR**-----> MAYA, UNREAL etc

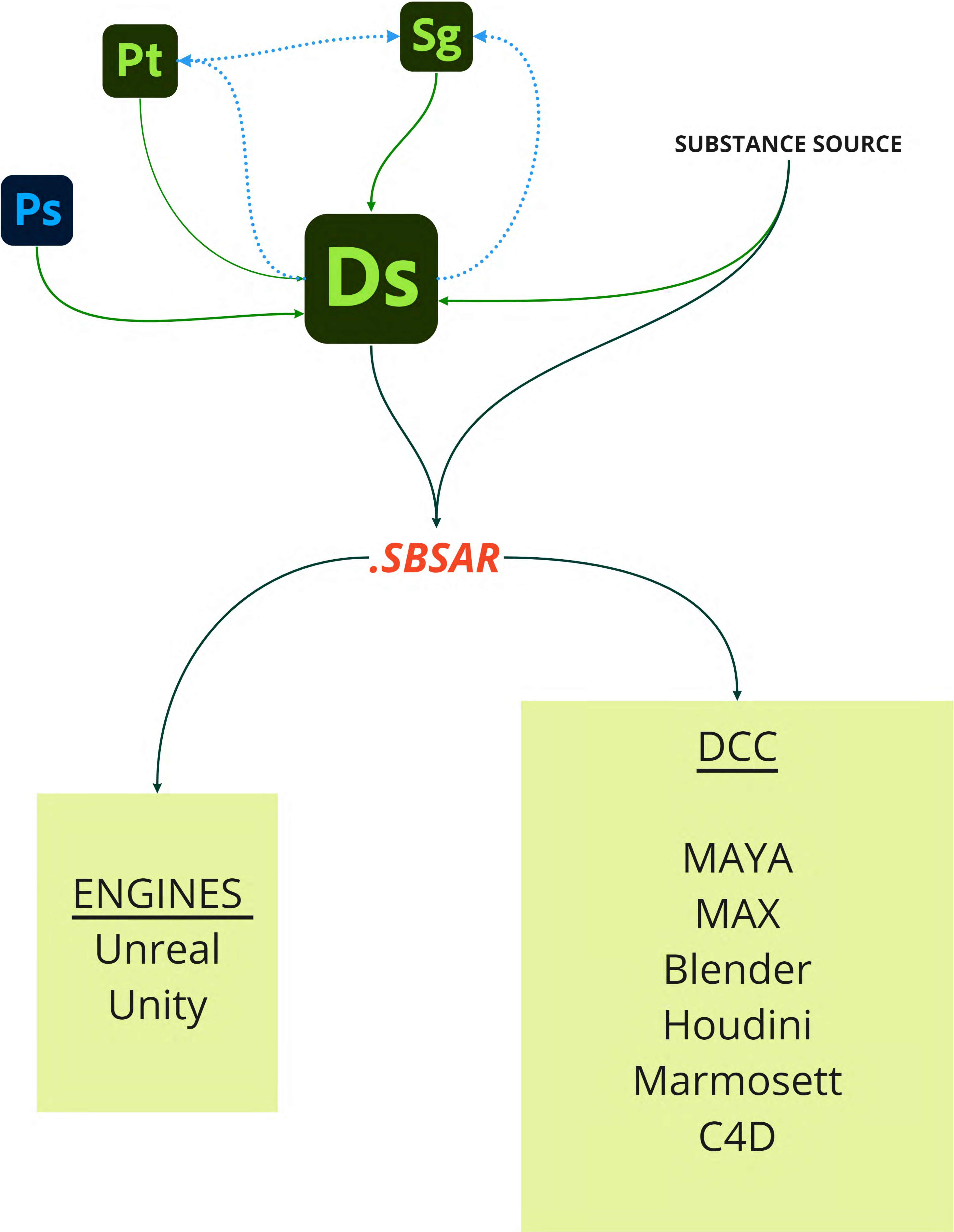
06 - UDIMS - One SHADER Compile for Everything. Each piece of Geometry uses a Udim space.

UDIMS MAPS
SBS - Shader 01



-----**.SBSAR**-----> MAYA, UNREAL etc

OVERVIEW



OVERVIEW

<https://substance3d.adobe.com/documentation/sddoc/pipeline-and-project-configuration-170459150.html>

[Publishing Substance 3D asset files \(SBSAR\)](#)

[Publishing concepts](#)

it is good to keep the following in mind when publishing a Substance compositing graph:

1. You publish a package, with all its contents, not an individual [Substance compositing graph](#). A Substance 3D asset then lets you generate content from all Substance compositing graphs inside this package.
2. Published packages are completely stand-alone: all resources required are embedded into the file. That means they are much easier to share than SBS files.
3. The output from Substance 3D assets can be completely dynamic. [Resolution is not set; exposed parameters can be modified](#). However, editing the Graph is no longer possible.
4. Substance 3D assets can be used outside of Designer, in all Adobe Substance 3D products, Adobe Dimension as well as any other application that has a [Substance integration](#).
5. Publishing is different from [Exporting](#), make sure you understand well the difference.

Preparing to publish

Publishing takes some more preparation than Exporting Bitmaps. That's because your published Substance 3D assets are dynamic tools, not just a static snapshot of the current state of your textures. Specifically, you want to keep the following in mind:

1. Make sure graph resolutions ([Output Size](#)) are set to the Relative To parent [inheritance method](#), which means they are dynamic and can be changed on the fly.
2. Make sure [graph outputs](#) are set up correctly with names, labels and usage tags.
3. Make sure [Parameters, if needed are organized and named properly](#).
4. Make sure the [Output size](#) property of all [Bitmap](#) nodes are set to the Absolute [inheritance method](#). If that is not the case, their referenced [Bitmap resource](#) will be saved at the default 256*256 resolution in the published Substance 3D asset file, which will impact the quality of one or more outputs.
5. If graphs are present in the package that should not be available outside of Designer (for example helper or "tool" sub-graphs that only work in a specific context), set them up to be hidden in their properties. See further below

OVERVIEW

Publishing methods

Once you are ready to publish, there are two ways to access the Publishing Dialog, both are through [the Explorer Window](#).

In the [Explorer window](#), Right-clicking on the package and choosing Publish .sbsar file..., alternative Hotkey Ctrl + P.

After Publishing with dialog once, you can also use Publish .sbsar file as previous to repeat the publishing process without seeing the dialogs, immediately publishing with the same settings.

In the [Explorer window](#), by clicking the Publish button in the top toolbar.

After Publishing with dialog once, you can also use the Publish as previous button to repeat the publishing process without seeing the dialogs, immediately publishing with the same settings.

Substance 3D asset limitations

While the Substance 3D asset is the most powerful and most dynamic format in the Substance Ecosystem, there are some small, technical limitations to be aware of.

1. Published Substance 3D asset packages are a one-way file format. You cannot "decompile" a Substance 3D asset back to a Substance 3D file (SBS). The only way to "edit" a Substance 3D asset, is to edit the original Substance 3D file. You can still use Substance 3D asset package contents as nodes inside new Substance compositing graphs (open and drag-and-drop), so this is not a huge limitation.
2. Substance 3D asset files have versions that infer compatibility. The core Substance Engine is updated from time to time with new features. packages that use these features need to be read by applications that support these new features. This is not an issue for all Substance Applications, as they are all updated at the same time, but Plugins and integrations might have longer compatibility delays.
3. Use the Substance Engine Compatibility display options in the [Project Preferences](#) to track down any potential issues.
4. Some exposed parameters do not function anymore once published as Substance 3D assets. See [Exposing a parameter](#) for a comprehensive list.

OVERVIEW

[Designer Pipeline setup](#)

Every type of file is explained with more detail on child pages of this page, but the short overview of how to ideally define a custom setup for Designer is as follows:

1. Identify and group what settings to add into your Project files. This is different for every studio and requires a certain amount of planning!
2. In almost every case at least 2 Projects should be defined: one for global, studio-wide defaults (like standard templates, shader files, baking settings), and one with more specific content such as Library content. If you have different projects running simultaneously you might want to create multiple project configurations for each (so 3 or more total).
3. Create the relevant [SBSPRJ files](#) and place them and their content under version control. It is strongly recommended to separate Designer pipeline and library content from your actual project content and resources (3D models, textures, code) by creating a separate repository for it.
4. Create an [SBSCFG configuration](#) file listing all project files, place it under version control. If you have multiple projects, you can create a Config for every project.
5. Set up every user's [User Preferences.xml](#) to reference their relevant configuration file.
6. You can have every user do this manually, or you can script this by injecting lines into their XML file. [More info on the relevant page](#).

[Project Configuration Files - SBSPRJ](#)

Project configuration files are the most complex and most expansive files used to configure Substance 3D Designer. They are special in that you can use multiple Project Configuration files, where each next "Child" Project expands or overrides the previous "parent". Unless explicitly needed, settings should not be modified or added to the Project files, so Designer can fall back on its parent configuration, or even defaults.

By default, Designer has 2 Project Configurations active:

1. Default Project
2. Contains all the default settings and the library Designer comes with on a fresh install. Read only, can not be modified or removed.
3. User Project
4. Because the Defaults are read-only, any changes by the user go into this project by default. Can not be removed.

This basic setup ensures that the default library and other settings can not be corrupted or modified, yet still allow single, amateur users to add their own modifications without having to bother with complex setups

OVERVIEW

[User Preferences - Automating Setup](#)

The `user_preferences.xml` file contains all user-specific settings outside of the ones defined in a [Project Configuration](#). These mainly relate to specific UI and performance settings.

The only relevant setting to change is the [Configuration File](#) that contains a list of projects.

This can be done in a few ways, as listed below.

Alternatively, you can completely bypass modifying the User Preferences, and do a session-based override of the SBSCFG file by using a command-line argument on the Designer shortcut, see below.

[Permanent or Session-based](#)

There are two different ways to configure Designer to use another [configuration file](#) than the default, both with advantages and disadvantages:

1. Permanently modifying `user_preferences.xml`
2. This file is located in `~User\AppData\Local\Adobe\Adobe Substance 3D Designer for Windows`. If you modify it, Designer will always use what is defined there, regardless of how, when or where you start it. Making changes requires modifying the XML again, both of which are described below, and tend to be a bit involved.
3. Temporarily setting the session through a command-line argument
4. Designer can take a command-line argument on startup to override the SBSCFG file for that session (see below for how). It's a simple, elegant solution, and allows for switching projects in a much quicker way than through modifying an XML. The danger is that if you open through multiple shortcuts (for example Start Menu and Desktop on Windows), you can have different results without it being entirely obvious. On top of that, it's not as tamper-proof, as users can delete, move or modify their shortcuts much easier than their `user_preferences.xml`

OVERVIEW

[Exporting models](#)

This page describes the process of exporting Substance Models from Substance Model graphs, and the available target file formats.

Overview

Once you have finished working on your Substance model graph, you can export the Scene output by the Root node to a file.

Exporting is performed by clicking RMB on the Substance model graph item in the Explorer panel, and selecting the Export... option. A system file browser opens, and you may select a destination location for the exported file, then set its name and type.

Depending on the type you select, the content of the Scene will be dynamic or static. The available types are presented below.

[Scripting](#)

It is possible to control the application by writing plugins. Plugins can be written in Python scripting language (based on version 3.7). The Python API provided in the application allows to access and use many features directly from scripts. Python is widely used scripting language by many industries. It helps for automation, workflow improvement tools and asset management for example. Learning materials are plentiful and readily available on the Internet (tutorials, commented projects, etc.). The API documentation can be accessed from the main menu with Help > Python API Documentation action.

LINKS ARE HERE :

1. [Plugin basics](#)
2. [Plugin search paths](#)
3. [Plugins packages](#)
4. [Plugin Manager](#)
5. [Python Editor](#)
6. [Accessing graphs and selections](#)
7. [Nodes and properties](#)
8. [Undo and redo](#)
9. [Application callbacks](#)
10. [Creating user interface elements](#)
11. [Adding actions to the Explorer toolbar](#)
12. [Using color management](#)
13. [Using spot colors](#)
14. [Logging](#)
15. [Using threads](#)
16. [Debugging plugins using Visual Studio Code](#)
17. [Porting previous plugins](#)
18. [Packaging plugins](#)
19. [Scripting API reference](#)