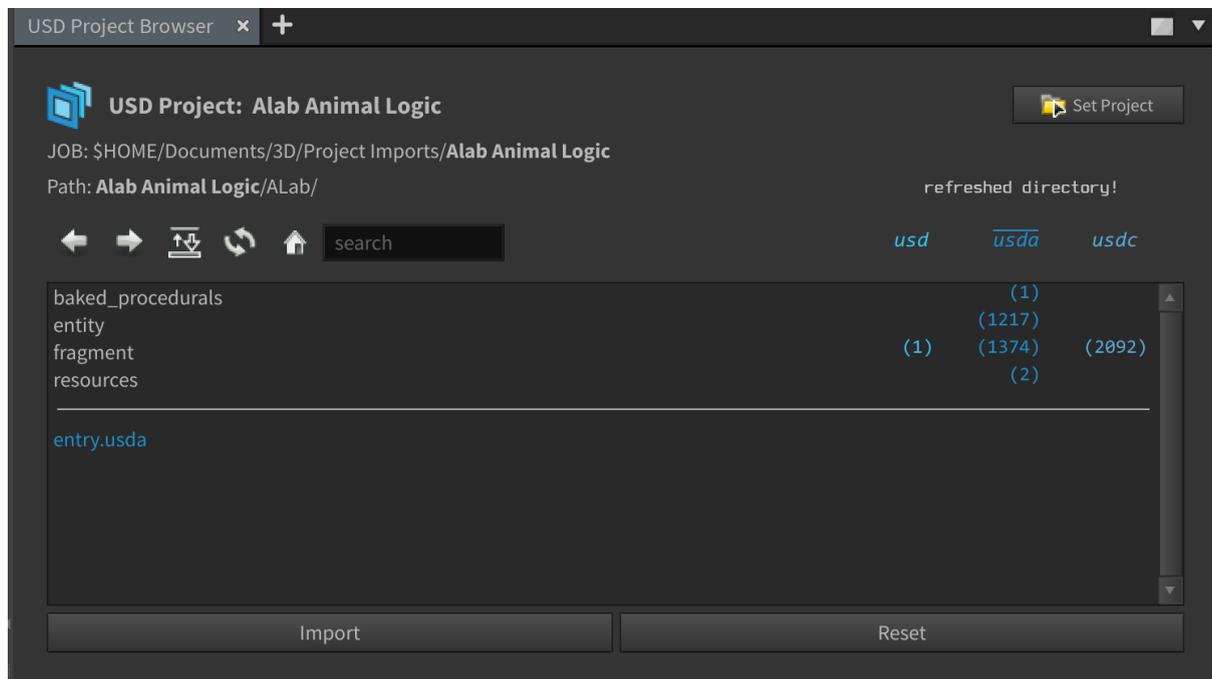


# Houdini USD Project Browser



## Index:

1. [Overview](#)
2. [Technical Details](#)
3. [Features and Functionality](#)
4. [Visual Demonstration](#)
5. [Integration and Compatibility](#)
6. [Version Control and Collaboration](#)
7. [Future Plans and Improvements](#)
8. [Source Code Access](#)

### 1. Overview:

This tool is a Houdini Pane Interface that acts as a File Directory Browser. It allows users to navigate selected directories to view and import USD related files. It provides an improved user experience versus a standard OS file directory by only focusing on and providing details for USD related files and directories. It also allows users to import USD files seamlessly by creating the geometry node `USD Import` with selected parameter settings in Houdini's network view.

### 2. Technical Details:

This tool was created with Houdini's Python Panel Editor that calls an external Python Script that utilises the os module, hou module (hou) and PySide2 module in conjunction with QtDesigner. The initial UI was created in QtDesigner and loaded into the Python script. PEP8 Style Guide Formatting was implemented in my code.

### 3. Features and Functionality:

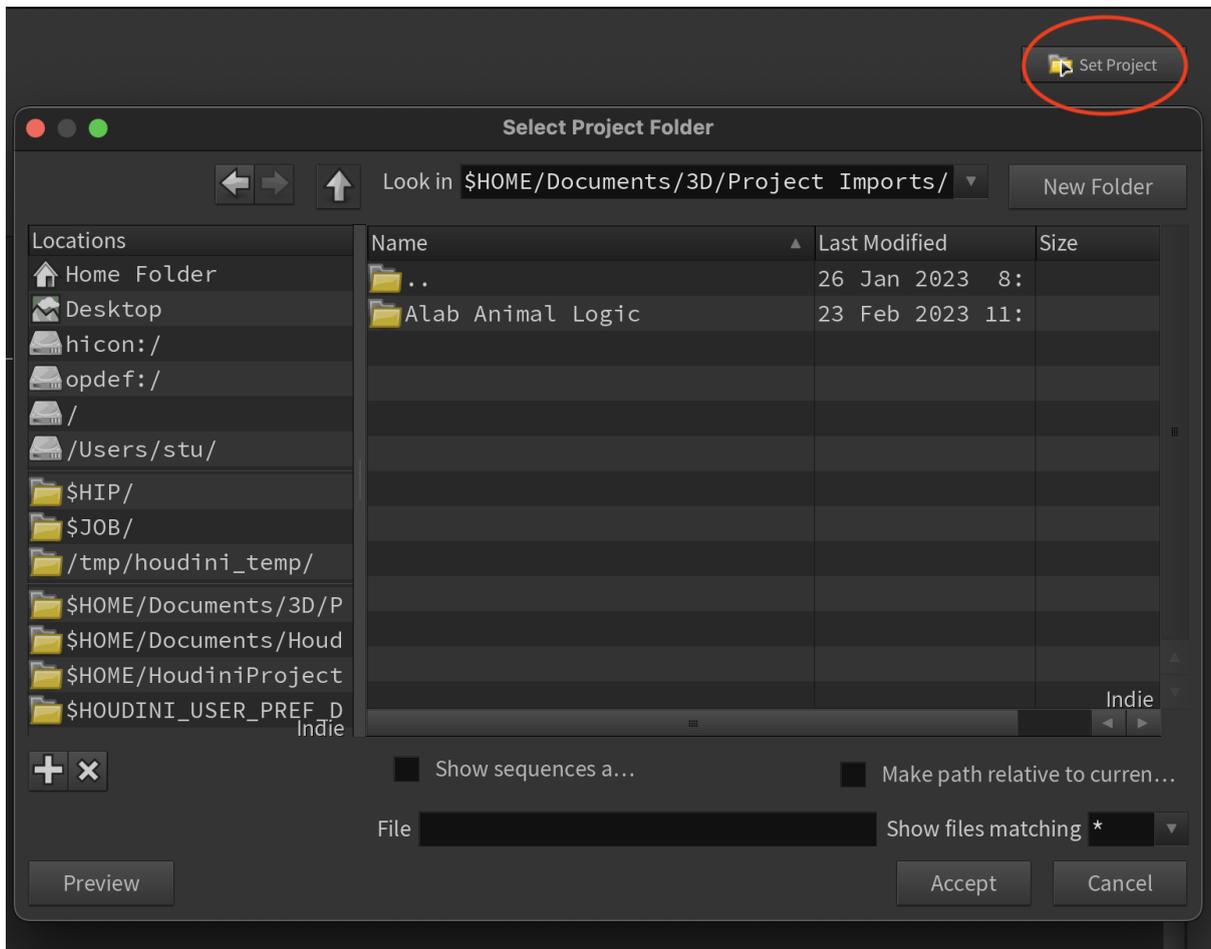
The features and functionalities of this tool are designed to make things easier for the user to keep track of USD related files in the production pipeline and to make an improved file directory user experience versus a standard OS file directory.

#### 3.1 Project Name Label



This label displays the name of the current project set by the user. This feature allows the user to easily keep track of what USD project the directory is set to as the functionality will adjust with each use of the 'Set Project Button'. The Pixar USD Logo enhances this feature.

#### 3.2 Set Project Button



This button acts as a typical Houdini Directory Chooser and as well sets the JOB variable to the directory. This feature kills two birds with one stone allowing the user to jump straight into a USD related project without worrying about setting the JOB variable beforehand.

---

### 3.3 Job Label

JOB: \$HOME/Documents/3D/Project Imports/Alab Animal Logic

This label displays the absolute file path of the JOB variable which is also the USD Project that the user has set. This feature allows the user to continually keep track of what the JOB variable has been set to and what its absolute path is in context to the user's operating system. The relative JOB file path at the end of the path is bolded to enhance the functionality of this feature.

---

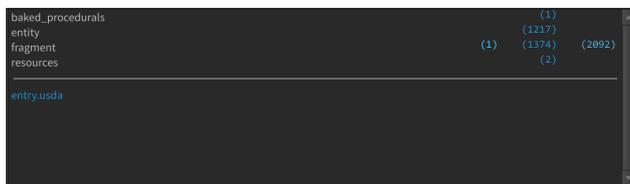
### 3.4 Path Label

Path: Alab Animal Logic/ALab/

This label displays the relative file path of the current path the user is navigating. Its functionality allows the label to adapt to each path navigation made by the user. The relative JOB file path at the start of the path is bolded to enhance the functionality of this feature.

---

### 3.5 File Directory List



This list item displays all directories and files listed within the currently selected directory. It features a scroll bar to the right and it has multiple types of functionalities linked by other features in the tool.

---

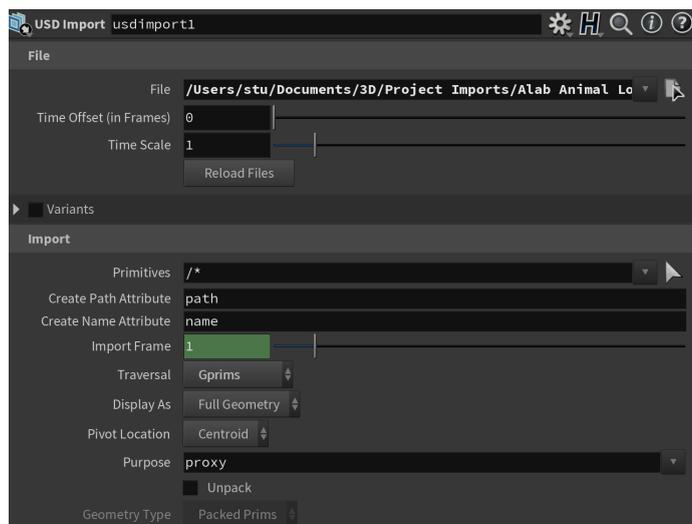
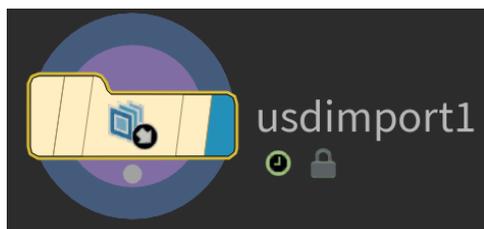
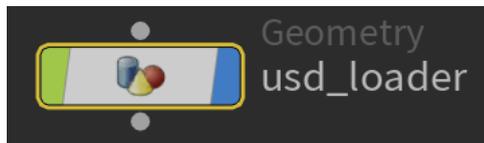
### 3.6 Import Button



This button allows the user to import usd type files into Houdini's network as geometry. A **`USD Import`** node is created within a **`Geometry`** Node as its functionality.

---

### 3.7 USD Import Node



When this node is created via the **`Import Button`** feature, the **`File`** parameter is set to the absolute path of the selected usd type file. The **`Traversal`** parameter is set to "Gprims".

---

### 3.8 Reset Button



This button allows the user to Reset the Browser tool back to its initial state, clearing the list and setting all buttons and labels to their initial default values. Upon pressing this button, a confirmation pop-up window appears unless deactivated. This stops the user from accidentally resetting the interface. This button is mimicking the “Reload the current interface” Houdini button.

---

### 3.9 Back Button



By pressing this button the user can navigate up one directory level without the need to select any item from the file directory list. This feature can also be activated by the user pressing the ‘left arrow key’ when the file directory list is in focus. The functionality of this button stops when the user is at the JOB directory.

---

### 3.10 Forward Button



This button allows the user to navigate down a selected subdirectory inside the file directory list. This feature can also be activated by the user pressing the ‘right arrow key’ when the file directory list is in focus and a directory is selected. The functionality of this button acts as a double feature, the second feature being **Forward Redo**.

---

### 3.11 Forward Redo



This feature serves as a ‘redo’ button for descending subdirectory navigations. It’s activated by the **Forward Button**, but only after using the **Back Button**, resembling traditional redo behaviour. It retains history, enabling users to redo until no path history remains or the bottom subdirectory is reached. No file directory list selection is needed, otherwise it functions like the standard **Forward Button**. Path history is cleared upon activating **refresh**, **home**, **reset**, or **set project** buttons and modified when users navigate different project directory paths.

---

### 3.12 Sort Button



This button sorts all directories in the file directory list alphabetically, toggling between ascending and descending order. By default, the Directory List is sorted alphabetically in ascending order. If a user sorts it in descending order and then navigates away and back, the list reverts to ascending order. This button only affects directory names and not user related files.

---

### 3.13 Refresh Button



This button refreshes the Project Directory, updating any additions or removals of files or directories made in an external file explorer, as well as resetting the currently viewed path to navigation defaults, which clears path history and restores default sorting. The user is notified of the update in the **Comment Label** feature.

---

### 3.14 Home Button

This button returns the user to the JOB directory path and resets the path to navigation defaults, clearing path history and restores default sorting. The user is notified of the update in the **Comment Label** feature. The **Path Label** will adjust accordingly.

---

### 3.15 Search Bar

This feature allows the user to search for any items within the currently viewed directory via a “substring search”.

Multiple hotkeys are functional in this feature including:

- “Escape” Key: First use clears the search bar content. Second use removes the selection of the search bar by the user.
  - “Enter” Key: Jumps to first available item from Search Bar onto file directory list
  - “Ctrl + Backspace” Key: Deletes all characters from the cursor position in the Search Bar.
- 

### 3.16 Comment Prompt

This feature displays a prompt to notify the behaviour of the program to the user. The use of certain features in the program will display specific prompts. An initial prompt label guiding the user is also displayed during the initial state of the program or when the program is reset.

---

### 3.17 USD Labels

This feature displays the name of each USD file type with their respective colour. If a USD item is present in the currently viewed directory, an overline will appear over the selected label. The overline indicates to the user that a USD file is located underneath the line separator of the current directory.

---

### 3.18 Mouse & Hotkey Navigation for File Directory List

source code

```
# Event handling methods
def mousePressEvent(self, event):
    if event.button() == QtCore.Qt.LeftButton:...
    self.enter_pressed_on_search_bar = False
    super(ProjectManager, self).mousePressEvent(event)

def keyPressEvent(self, event):

    if event.key() == QtCore.Qt.Key_Escape:...

    elif event.key() == QtCore.Qt.Key_Backspace:...

    elif event.key() == QtCore.Qt.Key_Delete:...

    elif event.key() == QtCore.Qt.Key_Return:...

    # Forward and back directory with arrow keys
    elif event.key() == QtCore.Qt.Key_Left:...
    elif event.key() == QtCore.Qt.Key_Right:...

    # Delete all text in search bar
    elif event.matches(QKeySequence("Ctrl+Backspace")) \
        and self.search_bar.hasFocus():...
```

This feature allows the use of multiple hotkeys & mouse click events for the user to navigate around the file directory list. These include:

- Navigation Arrows: If an item is selected in the list or an item has previously been selected in the list, the user can use the navigation arrows to navigate through the file directory. The back arrow acts as the **Back Button**. The forward arrow acts as the **Forward Button**. The up & down arrows allow the user to move vertically through items in the list.
- "Backspace Key": If an item is selected in the list, the backspace key deselects it.
- "Escape" Key: If an item is selected in the list, the escape key deselects it. If the escape key is used after the enter key has been used prior in the search bar, the escape key will then return the user's cursor back to the search bar, allowing the user to continue the chain of using the escape key to clear the search bar's content and exit the search bar all together.
- Left Mouse Button: If an item is selected in the list, clicking the left-mouse button outside of the list will deselect it.
- Double Clicking "Left Mouse" Button: If a directory is selected in the list and then double-clicked with the left mouse button, this feature will act as the **Forward Button**, creating the user to navigate down the selected subdirectory.

#### **4. Visual Demonstration:**

See reel - <https://vimeo.com/823980021?share=copy>

#### **5. Integration and Compatibility:**

Compatible with Houdini software. Tool is opened as a responsive window pane and can be torn off and resized.

#### **6. Version Control and Collaboration:**

<https://github.com/scmacoll/USDprojectbrowser>

#### **7. Future Plans and Improvements:**

- Create toggle button to set project without setting the JOB variable
- Enable “enter” hotkey to navigate down subdirectories in browser and to import usd files into the network view (with a initial notification window)
- Make all path label links clickable for path navigation
- Make USD labels clickable to sort USD files above or below the line separator
- Make sort button affect USD files above line separator
- Make compatible with Maya

#### **8. Source Code Access:**

Provided via email - [stuartcharlescreative@gmail.com](mailto:stuartcharlescreative@gmail.com)