Nuke– Watermark Util

Documentation

developer: Dushyant Kumar Kashyap

Date: 05/02/2016

Overview & Purpose

To create a tool that enables users to add watermarks on single images, sequences of images or rendered QuickTime movies.

Standards

1. Nuke 9.0v8.
2. Python 2.7.3 final

Objectives

1. Be able to process single imagery or multiple files with different file formats.
2. Be able to output each processing task into separate individual directories.
3. Allow users to choose output formats, movies/sequences output options, and output resolution.

4. Allow users to choose an image or to type in texts manually to use as watermark contents, and also allow users to create multiple watermark instances
5. Allow users to adjust the display of watermarks, positions, rotations, sizes and transparency levels

Files included

1. dushyant_info_watermark_code.py
2. dushyant_info_watermark_node.gizmo
3. watermark.png (for testing)

Installation

Steps to install the tool

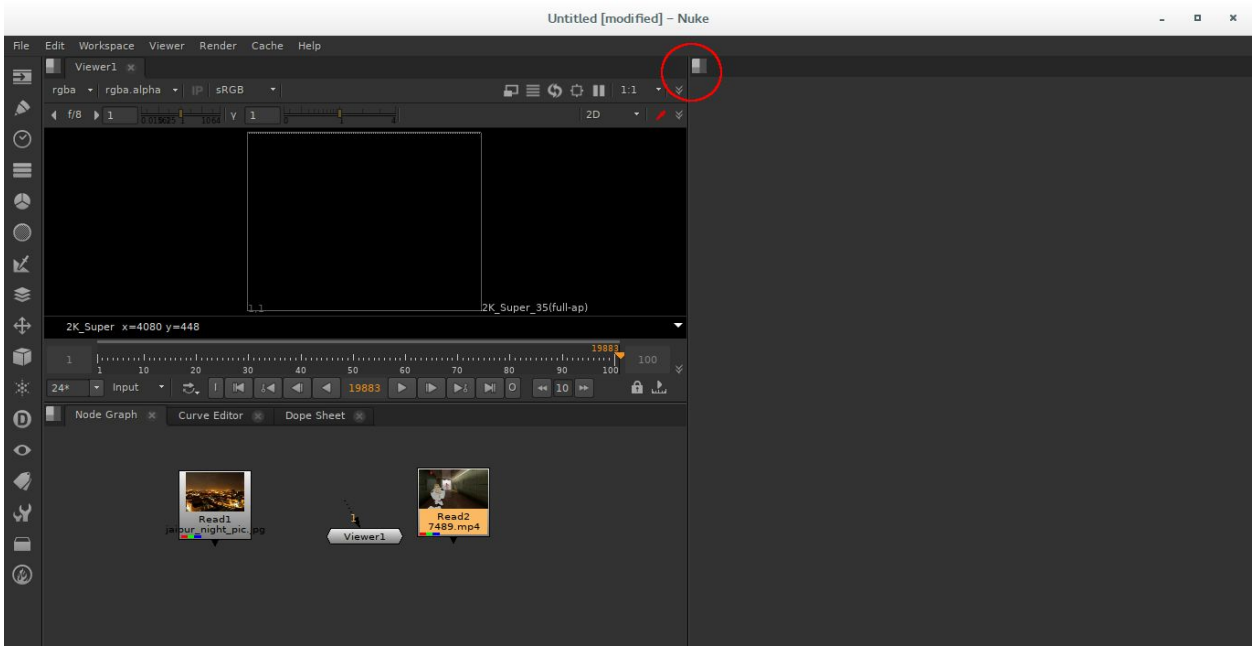
1. Download the files provided as attachment or you can clone https://github.com/dushyantk/nuke_watermark.git
2. Copy dushyant_info_watermark_code.py and dushyant_info_watermark_node.gizmo to your NUKE_PATH e.g. ~/.nuke/
3. In init.py (create one at ~/.nuke/ if it doesn't exists) add:

import dushyant_info_watermark_code at the end of the file.

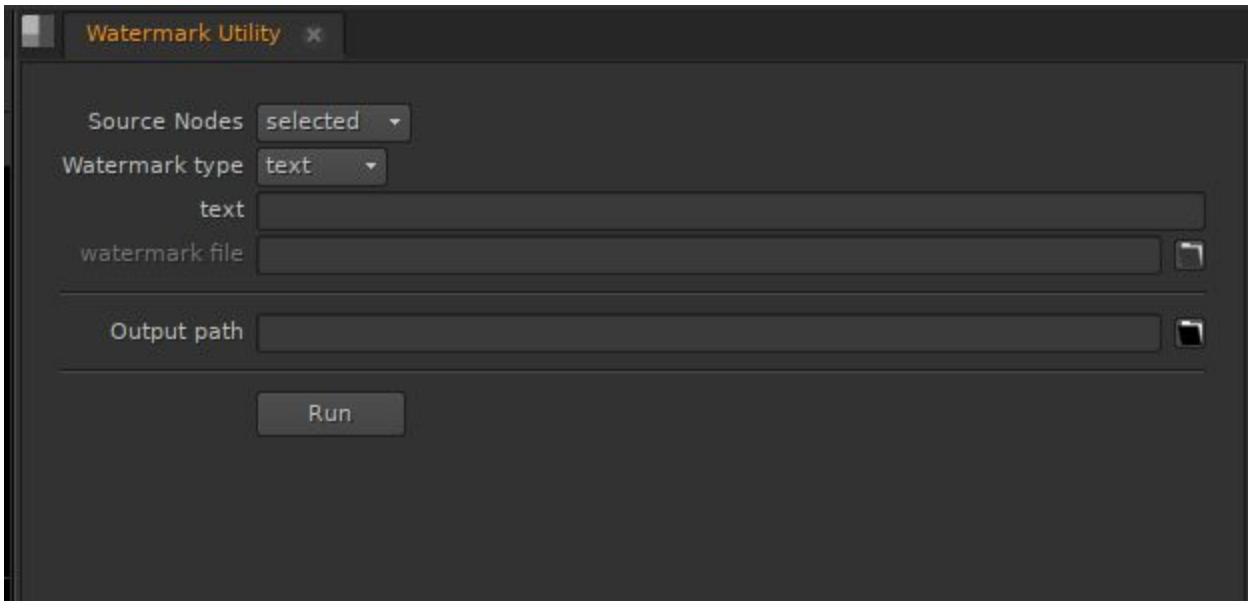
How to use (7 points)

Describe the process to use the tool

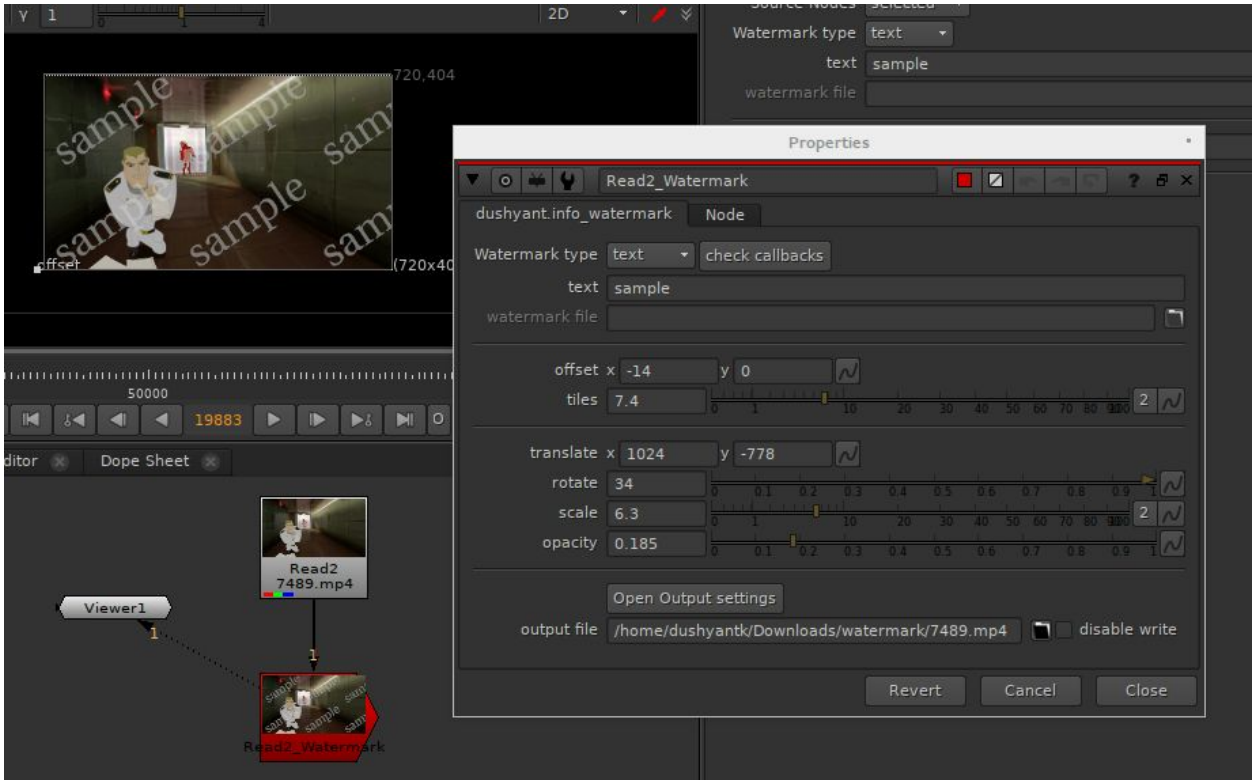
1. Click on pane menu in Nuke UI: (the area marked in red circle in the image below)



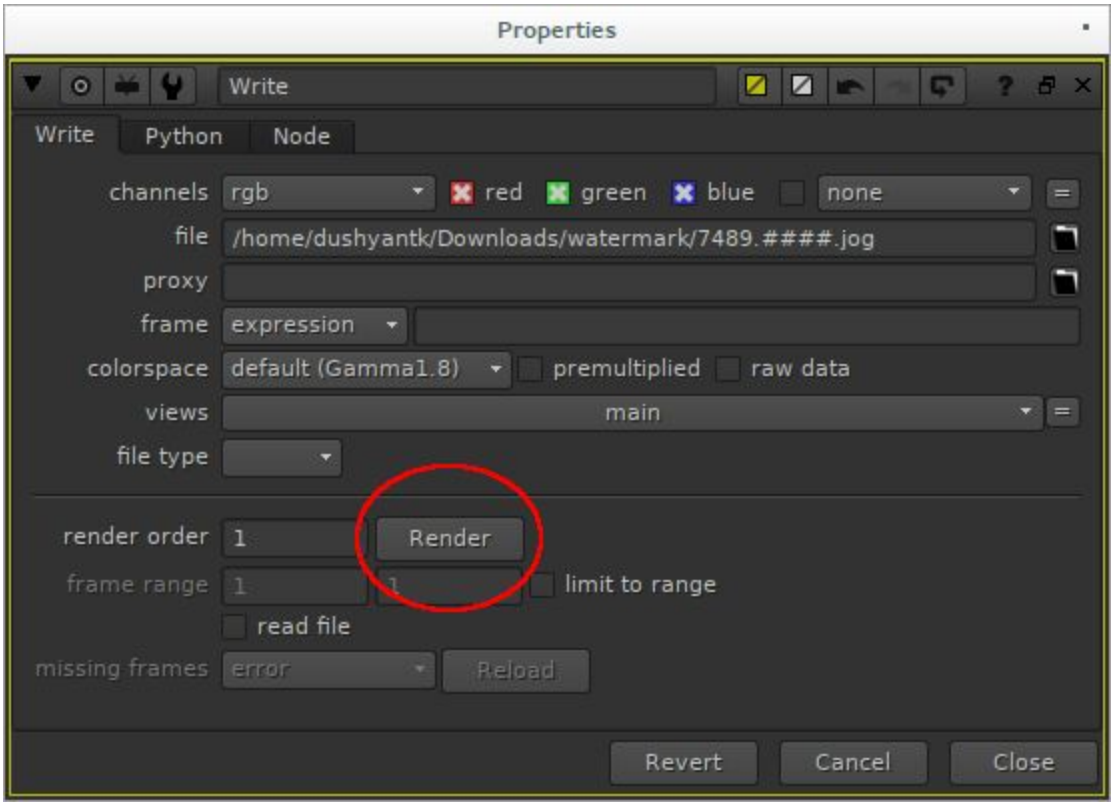
2. Then click on Windows > Custom > Watermark Utility
3. This will open Watermark Utility UI: I have chosen to add the UI as a pane because watermark creation seems like a repetitive job and the person doing it may want to save it in a layout and have it ready on each launch, panels in a pane allows you to do that, popup dialogs don't.



4. Here are the options described:
    a. Source Nodes (selected/all): you can select desired nodes (Read) to add watermark to or choose all to use all nodes in current nuke script. Github repo dushyantk/vfxPlumber/ (https://github.com/dushyantk/vfxPlumber/blob/master/python/utils/nuke/graphUtils.py) can be used to walk up the graph and condition @ line 235 (watermark can be attached to any node, not just Read.) can be skipped.
    b. Watermark type (text/image):
        i. text
        ii. watermark file, if image option is selected, you can use an image file as watermark.
    c. Output path: Contains the path to direct all the outputs, If this field is left blank, output path will be filled with source Read node's path with a directory named watermark on the same level, else this path will be used as the ROOT path for all outputs. **Note:** If you are leaving the *Output path* empty, make sure that you have read/write access to Read node's directory because the tool will try to create output directories during creation of the watermark node, and otherwise also, you should have correct permissions on the output path you are browsing.
    d. Run: Finish creating all watermark nodes.
5. Hitting "Run" will create a watermark node in your node graph attached to the read node selected:



6. Watermark node Options:
    a. Watermark type: this can be changed on the node itself to text or image.
    b. check callbacks: button for a TD to troubleshoot if text is not showing up or text/image switch is giving problems.
    c. text: text string to be used as watermark if image is selected as type.
    d. watermark file: path of the file to be used as watermark if image is used as type. A sample png with transparency is provided for testing.
    e. offset: offset watermark pixels from center, similar as translate.
    f. tiles: control the number of repetitions of watermark using this value.
    g. translate: translates the main watermark object.
    h. rotate: rotates the main watermark object.
    i. scale: scales the main watermark object.
    j. opacity: controls the opacity.
    k. open output settings: presents Write node settings to tweak and fire the the render.



l. output file: the automatically generated path of the output file.
m. disable write: **use this in case you want to attach your own Write nodes and bypass the one inside watermark node completely. Useful for batch processing on farm.**

7. The output directory was already created while generating the node, so you can hit render once you are satisfied with rest of the settings.