

[<< Back](#)

# Structures

## Drives

Real drives

Drive letter	Function
C:	Windows drive
D:	Sync work drive here + other work related stuff
E:	Sync all media here (foto, music, video)

Mapped drives

Drive letter	Function
I:	Active projects
M:	Music
P:	Personal files for software (settings, adons)

From the command line, subst [drive letter]: /d deletes the drive ===== *Music* ===== [Artwork finder](#) ===== *Folder structure* ===== \*MusicProduction \*Practice \*Ideas \*InProgress \*FinalTouches \*Finished \*Publish \*DJSets \*Library \*Samples \*AmbiguousRoyaltySounds \*FreeUse ===== *Photography* ===== *Folder structure* ===== \*Year \*01\_Incoming \*02\_Outgoing ===== *File structure* ===== [Template: YYYY-MM-DD\\_Location\\_Subject\(abbreviated, PubLib \(Public Library\)\)\\_FileVersion.Extention](#)

[Example: 2020-12-19\\_Zele\\_PubLib\\_001.jpg](#)

===== *CG* ===== *Folder structure* ===== \*YYYY-MM-DD\_NameClient\_NameProject \*\_Library \*00\_Incoming \*01\_Preproduction \*02\_Production \*01\_Preproduction \*01\_Moodboard \*02\_Storyboard \*02\_Production \*\_Blender \*\_DaVinci \*\_Houdini \*Scene\_01 \*03\_Outgoing \*YYYY-MM-DD ===== *File structure* ===== *Naming Convention* ===== [CamelCaps](#), geen spaties, underscores/streepjes enkel gebruiken bij opdeling filenaam.

Maak genoeg iteraties van je bestand, zodat je steeds een versie terug kan gaan, moest er iets mislopen.

==== *Models* ==== *Export files (FBX, OBJ)* hebben steeds zelfde versienummer als work file (Blender, Maya).

Afkortingen: \*Character: CH \*Prop: P \*Foilage: F [Template: Project\\_SceneNumber\\_Modeltype-ModelName\\_FileVersion.Extension](#)

[Example: Silver\\_000\\_CH-Nora\\_v001.blend](#)

==== *Texture Maps* ==== *Export files (PNG, JPG)* hebben steeds zelfde versienummer als work files (Substance, Photoshop) Afkortingen: \*Diffuse: DIFF \*Ambient Occlusion: AO \*Glossines: G \*Normal: N \*Roughness: R \*Subsurface Scattering: SSS \*Metallic: M [Template: Project\\_SceneNumber\\_TextureName\\_FileVersion\\_TextureType.Extension](#)

[Example: Silver\\_010\\_NoraHair\\_v001\\_DIFF.PNG](#)

[Example: Silver\\_010-NoraHair\\_v001\\_DIFF.PNG](#)

==== *Scenes* ==== *The count of scenes starts from 010, models with the scene number 000 are not restricted to one scene.*

[Template: Project\\_SceneNumber-Description\\_FileVersion.Extension](#)

[Example: Silver\\_010-MudWorld\\_v001.blend](#)

=== In 3D Software === Max 5 letters voor afkortingen

Afkortingen: \*Group: GRP \*Mesh: MESH \*Curve: CRVE \*Joint: JNT \*Light: LGHT \*Camera: CAM \*Locator: LOC \*Controller: CTRL \*Low Poly: LP \*High Poly: HP \*Forces: FRCE *Template: Type\_Name\_PolyType*  
*Example: GRP\_Nora\_LP*

===== Reference Library ===== For my collection of reference images, I only use broad categories as folders, the other information I put in the name and metadata. I'm using a combination of tools, including Advanced Renamer, digiKam, Python and a tool from GitHub called [taggui](#). == Workflow == First, I will change the name with Advanced Renamer.

Using TagGUI, I will then let the program generate captions into separate text files, which are named after the image they are describing.

Now we need the Python script to take the information from the text file, and add it to the metadata of the images.

For clean organisation, I opted to convert all files to .jpg. This to make sure the metadata is displayed in a clean way. == The name == CATEGORY\_SUBJECT\_inc Nr == The metadata == A description/caption in the images "Title" and "Subject" fields. == The script == 

```
code Python
[enable_line_numbers="true"]> import os from PIL import Image as PImage, PngImagePlugin from
pyexiv2 import Image as Pyexiv2Image def add_text_to_image_metadata(image_folder): try: # Iterate
through all files in the specified folder for filename in os.listdir(image_folder): image_path =
os.path.join(image_folder, filename) text_file_path = os.path.join(image_folder,
f"{os.path.splitext(filename)[0]}.txt") print(f"Processing {image_path}") # Debugging line if
os.path.exists(text_file_path): # Read the content from the text file with open(text_file_path, 'r') as
file: caption_text = file.read() # Convert filename to lowercase for extension checking filename_lower
= filename.lower() if filename_lower.endswith1): # Load the image's metadata image =
Pyexiv2Image(image_path) # Add the caption to the image's EXIF UserComment and
ImageDescription fields image.modify_exif({'Exif.Photo.UserComment': caption_text,
'Exif.Image.ImageDescription': caption_text}) elif filename_lower.endswith('.png'): # Convert the PNG
image to JPEG image = PImage.open(image_path) rgb_image = image.convert('RGB')
rgb_image.save(os.path.splitext(image_path)[0] + '.jpg', 'JPEG') # Load the image's metadata image
= Pyexiv2Image(os.path.splitext(image_path)[0] + '.jpg') # Add the caption to the image's EXIF
UserComment and ImageDescription fields image.modify_exif({'Exif.Photo.UserComment':
caption_text, 'Exif.Image.ImageDescription': caption_text}) # Delete the original PNG image
os.remove(image_path) # Delete the text file os.remove(text_file_path) print("Processing completed
for all images.") except Exception as e: print(f"Error: {e}") # Example usage: Specify the folder
containing your images image_folder_path = '/path/to/your/source/folder'
add_text_to_image_metadata(image_folder_path) }(); </code>
```

Sources: [https://www.youtube.com/watch?v=6KCtPnam6Sk&ab\\_channel=TimvanHelsdingen](https://www.youtube.com/watch?v=6KCtPnam6Sk&ab_channel=TimvanHelsdingen)  
<https://github.com/alexanderrichtertd/plex/wiki/Project-Structure>  
<https://www.youtube.com/watch?v=o2LqKH6ahDA>

1)

'.jpg', '.jpeg'

From:

<https://floriandheer.com/brainii/> - **Brain II**

Permanent link:

<https://floriandheer.com/brainii/doku.php?id=start:structures&rev=1722415676>

Last update: **2024/07/31 10:47**

