3D-Coat AppLinks specifications

6 april 2018

Applicable to 4.8.15 and later

AppLinks are intended to simplify data exchange between 3D-Coat and other applications. AppLink is a plugin written for the specific application.

1. The exchange path.

In your plugin you should compose path to exchange data. It is

My Documents\AppLinks\3D-Coat\Exchange

You need full path, so it will look like (but not exactly, it differs from PC to PC)

C:/Users/UserName/Documents/AppLinks/3D-Coat/Exchange/

See the Application 3 how to get this path. Other way is to ask this path from user, but it will be less convenient. Best option is to fill it by default and let user possibility to change it.

Let us call this full path **exchange_path.**

2. List of AppLinks in File->Export To->

In the **exchange_path** you need to create Folder that corresponds to your application, for example Blender. Like

exchange_path/Blender

Each folder means that there will appear the item in File->Export To->
Within the folder you should create files:

run.txt
The file is mandatory, it should exist. It may be empty, but if there is path to executable file, it will be run each time you choose in **3D-Coat File->Export To->...your app...**

extension.txt
Contains extension for export. If file not exists, OBJ is taken.

params.txt
Parameters that passed to executable from run.txt except path to object

preset.txt
The name of export preset that will be used for export

***.xml**
Any xml file in this folder will be treated as export preset. It will be copied to presets folder and used as default preset for this applink. You may refer this preset as file name without extension.

3. First scenario: Application -> 3D-Coat and back

There are several scenarios of Applinks usage:
First scenario: sending object from Application (say Blender) to 3D-Coat and back.

Next steps. Let user created some model in Application, wants to drop it to 3D-Coat.

First, as soon as user wants to drop model to 3D-Coat from the Application, you need to export object from Application, say the full path is

exchange_path/exportfile.obj

You expect that after all job over the model 3d-coat will create file with path **exchange_path/returnpath.obj** (generally name may be any different, this name is just for instance)
Create **import.txt** like

exchange_path/exportfile.obj
exchange_path/returnpath.obj
[ppp]
[export_preset Blender Cycles]

[ppp] means that you need to open file for perpixel painting. There are a lot of other options how to open the file. See the list below:

- Paint mesh in 3D-Coat using per-pixel painting **[ppp]**
- Paint mesh in 3D-Coat using microvertex painting **[mv]**
- Paint mesh in 3D-Coat using Ptex **[ptex]**
- Perform UV-mapping in 3D-Coat **[uv]**
- Drop reference mesh to 3D-Coat **[ref]**
- Drop retopo mesh as new layer in 3D-Coat **[retopo]**
- Drop mesh in 3D-Coat as voxel object **[vox]**
- Drop mesh in 3D-Coat as single voxel object, all objects will be merged together in one volume **[voxcombine]**
- Drop mesh in 3D-Coat as new pen alpha **[alpha]**
- Drop mesh in 3D-Coat as new merging primitive for voxels **[prim]**
- Drop mesh in 3D-Coat as a curve profile **[curv]**
- Drop mesh in 3D-Coat for Auto-retopology **[autopo]**

The last line ***[export_preset Blender Cycles]*** means that you will choose **Blender Cycles** preset in export dialog when user will export. Generally you may refer to any preset, even to your own.
You may place additional commands into import.txt. There is one useful example. Say, you need to run some script after importing the object. In this case write after **[ppp]** or **[export_preset ...]**

[script text_of_script]

or

[scriptfile path_to_script_file]

In the last case file referred as **path_to_script_file** should contain **main(){...}**
Example of useful script:
For example, after importing object into 3D-Coat you need to pass textures additionally. In this case write

[script ImportTexture("$LOADTEX","Uv_Set_Name", “Path_ToTexture”);]

"$LOADTEX" is identifier from Textures->Import->Color/Albedo map. You may take any identifier from **Import/Export menu**. Use **MMB+RM**B to extract identifier (it always starts from **$** sign).
Other examples of identifiers

$LOADTEX - Textures->Import->Color/Albedo
$ExternalAO - Textures->Import->External AO
$ExternalCavity - External Curvature

$LOADMETAL - Textures->Import->Metalness Map
$LOADEMISSIVE - Textures->Import->Emissive
$LOADEMISSIVE_BW - Textures->Import->Emissive Intensity
$LOADLOPOLYTANG - Textures->Import->Normal Map
$ImportOSNormal - Textures->Import->World Space Normal Map
$LOAD_LAYER_DISP - Textures->Import->Displacement Map

In **Roughress/Metallness** workflow:

$LOADROUGHNESS - Textures->Import->Roughness

In **Gloss/Metallnes**s workflow:

$LOADSPEC - Textures->Import->Glossiness Map

In **Gloss/Specular** color workflow:

$LOADSPECCOL - Textures->Import->Specular Color

If you need to switch to specific texturing workflow, use any of commands before textures importing commands

[script cmd("$GlossSpecular");]
//Textures->Textures Export/Import workflow->Gloss/Color Specular

[script cmd("$GlossMetallness");]
//Textures->Textures Export/Import workflow->Gloss/Metalness

[script cmd("$RoughnessMetallness");]
//Textures->Textures Export/Import workflow->Roughness/Metalness

Let us continue describing other parameters of **ImportTexture**

Uv_Set_Name is name of Uv-set where you want to import texture. Passing **"any"** means first UV-set of the model.

Path_ToTexture is full path to texture. Generally relative paths allowed as well, but path should be relative to MyDocyuments/3D-CoatVxx/. **Important!** It is better to pass all paths with **"/"** not with **"\"**. You may use **"\"** but you should pass **"\\"** instead of **"\"** due to **c++ syntax** used for scripts.
This is just example, you may use any script commands there.

There are several additional commands that you may use in import.txt, the list is in **Application 6.**

Important! You should create all object files, textures and only after all create **import.txt** because import.txt creation is signal for 3D-Coat to start importing.

After creation of the file import.txt 3D-Coat will import your file. In the case above - for perpixel painting.
Then user will edit the file, create textures, modify etc. After the edit process user will trigger the

File->Open in original App

In this case 3D-Coat will create set of files that will indicate that export finished. The file

exchange_path/export.txt

will be created. The file contains path to exported model file. It is the same name that you passed previously using **import.txt**. In our case it is

exchange_path/exportfile.obj

Also the file **exchange_path/textures.txt** will be created. It contains references to textures created during export

Name_of_material_1
Name_of_uv_set_1
Texure_uage_1
Absolute_path_to_texture_1
….same for second and other textures …

Texture usage is taken from the export presets tags, so you may easily identify type of texture. If Export constructor is not used then tags **color**, **specular**, **displacement**, **normalmap** are used**.** It there is tag **displacement** additional floating number will be written below – scale factor for displacement, it looks like

displacement 1.345

When you will detect that file export.txt exists and is not empty or locked - read the file, use it's content and delete. The file export.xml will be created as well. It contains complete export settings. No need to use this file if you don't need to analyse export settings.

So, you finished complete loop Application->3D-Coat->Application

4. Second scenario: 3D-Coat -> Application

There is menu item **File->Export To->....list of applications…** in 3D-Coat. In the **Section 2** we described how to add items there.

Afterl user will choose **File - > Open in original App** the file **exchange_path/YourAppName/export.txt** will be created. **You should monitor when the file will appear** and then read it. The **export.txt** contains the path of the object to import. After reading you need to delete the export.txt. Together with this file the **exchange_path/textures.txt** will be created. You may read the list of exported textures as described above. Pay attention that **export.txt** and **textures.txt** located in different folders.

5. Examples

Example1:

User pressed ***"Paint mesh in 3D-Coat using per-pixel painting [ppp]"*** in your AppLink

Let **exchange_path** is **c:/Users/Andrew/Documents/AppLinks/3D-Coat/Exchange/**

place some **OBJ/LWO/FBX** file there like **sample.obj** (mtl and textures too if need)
create file **c:/Users/Andrew/Documents/AppLinks/3D-Coat/Exchange/import.txt** and place 3 lines there:
[look the end of document how to locate the folder for OSX/Linux]

c:/Users/Andrew/Documents/AppLinks/3D-Coat/Exchange/*sample.obj*
c:/Users/Andrew/Documents/AppLinks/3D-Coat/Exchange/*output.obj*
[ppp]
[export_preset Blender Cycles]

Note that you may drop multiple objects simultaneously, you should separate them with ; sign. They will be merged together before import. There should not be extra spaces between names and ; signs. Example:

c:/Users/Andrew/Documents/AppLinks/3D-Coat/Exchange/*sample1.obj;*c:/Users/Andrew/Documents/AppLinks/3D-Coat/Exchange/*sample2.obj*
c:/Users/Andrew/Documents/AppLinks/3D-Coat/Exchange/*output.obj*
[ppp]

[export_preset Blender Cycles]

Once you will save file 3D-Coat will show import dialog for per-pixel painting and delete file import.txt. Press OK, draw something. Then use File->Bring object back (Open in original app). 3D-Coat will store file

c:/Users/Andrew/Documents/AppLinks/3D-Coat/Exchange/export.txt

This file contains full path to the output object. This is the same name as was specified in import.txt, so you will get one line in this file:

c:/Users/Andrew/Documents/AppLinks/3D-Coat/Exchange/*output.obj*

Also 3D-Coat creates file
c:/Users/Andrew/Documents/AppLinks/3D-Coat/Exchange/textures.txt

With lines:

cube1_auv
cube1_auv
diffuse
c:\Users\Andrew\Documents\AppLinks\3D-Coat\Exchange\output_diffuse.tga
cube1_auv
cube1_auv
reflection
c:\Users\Andrew\Documents\AppLinks\3D-Coat\Exchange\output_reflection.tga
cube1_auv
cube1_auv
roughness
c:\Users\Andrew\Documents\AppLinks\3D-Coat\Exchange\output_roughness.tga
cube1_auv
cube1_auv
normal_map
c:\Users\Andrew\Documents\AppLinks\3D-Coat\Exchange\output_normal_map.tga
cube1_auv
cube1_auv
emissive
c:\Users\Andrew\Documents\AppLinks\3D-Coat\Exchange\output_emissive.tga

Creation of the file export.txt is signal to other application to import files and replace object in scene by the object that was exported from 3D-Coat.
Your plugin should delete file export.txt after reading it.

Note: Objects may contain several sub-objects, several materials and several uv-sets. UV-set may contain several materials. You should handle this general situation.

Appliction 1. Seeking for exchange folder for OSX

common function for all OS:

```
void AppendSlash(char* s){
   int L=strlen(s);
   if(L){
      char c=s[L-1];
      if(c!='\\' && c!='/')strcat(s,"/");
   }
}
```

```
char str[512];
FSRef F;
if(noErr == FSFindFolder(kUserDomain, kCurrentUserFolderType, kCreateFolder, &F)) {
      FSRefMakePath(&F, (unsigned char *)str,512);
      AppendSlash(str);
      strcat(str,"AppLinks/3D-Coat/Exchange/");
 }
```

Application 2. Seeking for exchange folder for Linux

```
char str[512];
strcpy(str,g_get_home_dir());
 AppendSlash(str);
strcat(str," AppLinks/3D-Coat/Exchange/");
```

Application 3. Seeking for exchange folder for Windows

```
TCHAR Path[MAX_PATH];
if(SUCCEEDED(SHGetFolderPath(NULL, CSIDL_PERSONAL | CSIDL_FLAG_CREATE, NULL, 0, Path))) {
   AppendSlash(Path);
   strcat(Path, "AppLinks/3D-Coat/Exchange/");
}
```

Application 4: The general supposed architecture of plugin

Initialisation section:

- Get and store path of Exchange folder
- Create folder **Exchange/YourApp/** and file **run.txt** [must] and **params.txt** [optional]
- Create timer/thread to check if file import.txt exists. If threads or timer is not available, skip this step.

Processing section:

- Once in 1 sec you should check if any of files **Exchange/import.txt** and **Exchange/YourApp/import.txt** exist. If plugin has no timer callback or threads it has some processing cycle anyway. In this processing cycle you may check if 1 sec passed since last check of files.

It is supposed that plugin should not run 3D-Coat because there are many different versions and it will be difficult to determine what version user wants to use. But it will be helpful (but not required, optional) if plugin will check if process 3D-Coat is in memory and suppose to user to run it if process is not run. You should seek process that has substring “3D-Coat” in it’s name. There is sample of function to determine if 3D-Coat is in memory:

```
#include <tlhelp32.h>

bool Find3DCoat(){
       HANDLE hSnapshot = CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, 0);
       PROCESSENTRY32 pe;
       if(!hSnapshot)return false;
       pe.dwSize = sizeof(pe);
       for(int i = Process32First(hSnapshot, &pe); i; i=Process32Next(hSnapshot, &pe)){
              HANDLE hModuleSnap = NULL;
              MODULEENTRY32 me;
              hModuleSnap = CreateToolhelp32Snapshot(TH32CS_SNAPMODULE,pe.th32ProcessID);
```

```
		if(hModuleSnap == (HANDLE) -1)continue;
		me.dwSize = sizeof(MODULEENTRY32);
		if(Module32First(hModuleSnap, &me)){
			do{
				char temp[MAX_PATH];
				strcpy_s(temp,MAX_PATH,me.szExePath);
				_strupr_s(temp,MAX_PATH);
				int p=strlen(temp);
				char c=0;
				while(c!='\\' && c!='/' && p!=0){
					c=temp[p--];
				}
				char* s=temp+p;
				strupr(s);
				if(strstr(s,"3D-COAT")){
					return true;
				}
				break;
			}while(Module32Next(hModuleSnap, &me));
		}
	}
	CloseHandle(hSnapshot);
	return false;
}
```

Application 5: Possible problems while making plugin.

- If you decided to seek if 3D-Coat's process is in memory and run 3D-Coat (this step is optional) you need to run 3D-Coat as administrator on Vista/7
- You are specifying the export path and file name for 3D-Coat in import.txt. If you are specifying the same path to the output object for different scenes it can cause problem because textures will be overwritten with textures from the new scene. You should specify path that will depend on user's scene/name of object or so to avoid this problem.

Application 6: Extra commands for **import.txt**

Import.txt may have additional options in additional lines of the file.

- [SkipImport] allows to skip import dialog, default options will be used.
- [SkipExport] allows to skip export dialog. Previous settings will be used (works in **4.8.16** and later)
- [TexOutput:… texture path there…] allows to specify textures output path. It should end with "\" or "/". **This option does not work with export constructor**. When export constructor used, textures paths are determined by object export path.

Set of export options could be changed using command like
[Option=value]
There is set of possible options and values:

Option	***Value***	***Works for export constructor?***
ExportColor	0 or 1	No
ExportSpecular	0 or 1	No
ExportNormalmap	0 or 1	No
ExportDispl	0 or 1	No
PickSourcePositions	0 or 1	**Yes**
ExportSpecularColor	0 or 1	No
ExportEmissive	0 or 1	No
ExportEmissivePower	0 or 1	No
ExportMetallness	0 or 1	No
ExportRoughness	0 or 1	No
ExportAO	0 or 1	No
PickDepthFromLayer0	0 or 1	No

CoarseMesh	0 or 1	No
DisplDepth	8 bits, 16 bits, 32 bits	No
DisplNorm	GREYBASED, ZEROBASED, ZEROBASED_NORM, ZEROBASED_ABS	No
DisplExt	BMP,TGA,PNG for 8 bit displacement TIF for 16 bit displacament TIFF, EXR for 32 bit displacement	No
ColorFileExtension	TGA, BMP, PNG, JPG**, TIF, TIFF, EXR, PSD**	No
SpecNormExtension	TGA, BMP, PNG, JPG**, TIF, TIFF, EXR, PSD (only specular extension will be assigned with this command, normalmap is TGA by default unless specified directly)**	No
SpecExtension	**TGA, BMP, PNG, JPG, TIF, TIFF, EXR,PSD**	No
NormExtension	**TGA, BMP, PNG, JPG, TIF, TIFF, EXR**	No
ExportResolution	LOW-POLY, MID-POLY	**Yes**
[field field_id=value]	**Set field value in export dialog. This command generally replaces all of previous commandes. You may set any field in export dialog. Example [field $ExportOpt::UseExportConstructor = true] To get field_id click LMB+RMB or MMB+RMB over the required field in export dialog.**	**Yes**
[click control_id]	**Click on any control in export dialog. Example - choose Unity preset - [click $COMBOBOX_Unity (Specular)] To get field_id click LMB+RMB or MMB+RMB over the required field in export dialog.**	**Yes**

Generally it is much better to operate using export presets, way through “Options” is deprecated.