vollken_uploader

A Small tool to upload joined file to vollken_service

Forms:

- GUI
- CLI

Description

This tool will allow only to upload joined files to our S3 repository. The uploaded files will be automatically processed via our cloud distributed solution `vollken_service`

Configuration

The tool looks for the `./config/vollken_uploader.cfg` file for the configurations.
There's a sample configuration file `vollken_uploader.cfg.sample` that can be copied and used as the starting point for the configurations.
Alternatively you can make a similar file with `[vollken_uploader]`section holding with these parameters:

```
[vollken_uploader]
username=************
password=************
timezone=America/Los_Angeles
```

Usage

GUI: The usage in the GUI tool is hoepfully self-explanatory.
CLI: The CLI tool must have only 1 argument -i/--input, which will be the joined file.

CLI options:

```
-i/--input: local input file (joined.mvx)
-g/--local-graph: Local graph to upload (optional)
-b/--build: Custom build to use (optional)
-n/--num-slices: Number of instances to launch (default: 15)
--rclone: Using rclone as an uploader (default)
--aws-cli: Using aws-cli as an uploader
--boto3: Using boto3 as an uploader
```

NOTE: When using the `--build` command, don't specify the extension.

Examples:

```
PS> .\VollkenUploader.exe -i
"D:\devRing\studio\27_frames_dance\StudioDance_joined.mvx" --boto3
```

In this example the UUID will be `StudioDance` (case-sensitive).

More complex:

```
PS> .\VollkenUploader.exe -i
"D:\devRing\studio\27_frames_dance\StudioDance_joined.mvx" --build v2011 --
local-graph "D:\graphs\new_HQ_mesh.xml" --aws-cli
```

Upload Method

This tool has 3 uploding methods, and can be defined with a CLI param as above (boto3 doesn't need to be configured manually).
In order to use either rclone or aws-cli, one must install it's binraies and configure them properly.

AWS-CLI-V2

aws-cli can be download from here. Then configure with the VollkenUploader credentials (provided by Mantis-Vision)

```
PS> aws configure set aws_access_key_id ************
PS> aws configure set aws_secret_access_key  ************
PS> aws configure set region  ************
```

Rclone

rclone can be download from here. Then configure with the VollkenUploader credentials (provided by Mantis-Vision)
Use `rclone config` command with a new remote named **`s3`**.

Metadata file

Each joined file must have a metadata JSON file, including information regariding the asset. The gile must be in the same directory as the joined file holding the same name except the suffix `_metadata.json`.
This metadata file must be in this format exactly, if not the tool will raise an excpetion.
NOTE: the *_URI attributes are empty for a reason, so keep them that way.

```
[ <- Note the brackets
  {
    "AccountID": "<STRING>",
    "groups": "<STRING>",
    "tags": "<STRING>",
    "UUID": "<STRING>",
    "Title": "<STRING>",
    "Description": "<STRING>",
    "Privacy": "True | False",
    "VID_URI": "",
    "DATA_URI": "",
    "THUMBNAIL_URI": "",
    "Quality": "HQ | RT",
    "email": "<STRING>",
    "Date": "<STRING %m/%d/%Y %H:%M:%S format>"
  }
]
```

Audio files

Each joined file can have audio files (`.wav` and `.ts`), which will be uploaded also and later will be merged by `vollken-service`.
For a successful merge, both files must be present, but `vollken-uploader` wiil not raise any Exception or error upon missing either of them.

Input directory example

```
.
├── <UUID>_joined.mvx
├── <UUID>_joined.xml
├── <UUID>_metadata.json
├── <UUID>.ts  (optional)
└── <UUID>.wav (optional)
```

Limitations and conventions:

- Asset UUID will be extracted from the MVX joined file name before the "_joined.mvx".
- Do not use underscore (_) in the name at all.
- Each MVX joined file must end with "_joined.mvx".
- Each MVX joined must have a metadata file.
- Metadata file name must be with the same name as the joined file (without the suffix).