Setting up the media server on a Windows /vg/station--for people who aren't the /vg/ coders (7/20/21)

For players:

1. Download 32-bit VLC from https://www.videolan.org/vlc/#download (check the URL for a "win32.exe" ending, example: https://get.videolan.org/vlc/3.0.16/win32/vlc-3.0.16-win32.exe). If you have a 64-bit VLC installed, uninstall that first.
BYOND is 32-bit and thus requires browser plugins that can run on 32-bit only.
2. When prompted, ensure **ActiveX plugin** is checked under the components.
3. Proceed with regular installation.
4. Ensure ActiveX is enabled on Internet Explorer. If you are on Windows 11, use your search to look for **Internet Options**.
    5. On the **Security** tab, click **Custom level…**
    6. Scroll down to **ActiveX controls and plug-ins**.
    7. Under **Allow ActiveX Filtering**, click **Disable**.
    8. Click **OK** at the bottom of the dialog.
    9. Click **Apply** on Internet Options, then **OK**.

For host:

Thankfully (?), the "media server" is entirely written in PHP and is thus capable of running (?) on Windows.
Configuring and running the media server on Windows
1. Download and install XAMPP on your VM.
    https://www.apachefriends.org/download.html
2. Start the XAMPP control panel program.
3. Open the XAMPP server root (by default, this is `C:/xampp`)
4. Delete `htdocs`. This folder will be replaced by the `htdocs` folder from the Media Server.
5. Using git or zip download, clone the Media Server folder (SS13-media) somewhere on your VM.

    `git clone https://github.com/vultraz168/SS13-media.git`
    OR download:
    https://github.com/vultraz168/SS13-media/archive/refs/heads/master.zip

6. From `SS13-media` copy and paste `htdocs`, `lib`, `cache.php` and `playlists.json` to your server root.
7. Create a folder called `cache` in your server root.
8. Create a folder called `files` in your server root.
9. From SS13-media copy and paste `config.php.dist` into your server root and rename it `config.php`
11. Open `config.php` with a text editor.
12. Set the value of ROOT_URL to "http://" + your externally visible ip (not localhost or 127.0.0.1) + ":80"
For example, if your VM IP is 3.238.18.42, the config should say:
    `const ROOT_URL = "http://3.238.18.42:80";`
13. Optional: Set the value of API_KEY to anything you want or to `''` to not require any key (can be useful for debugging).

This acts as a password to prevent people without the API key from easily accessing the playlists/songs on the media server outside of BYOND.

14. In the XAMPP control panel, click the first **Start** button (the one next to **Apache**). This will start the media server.
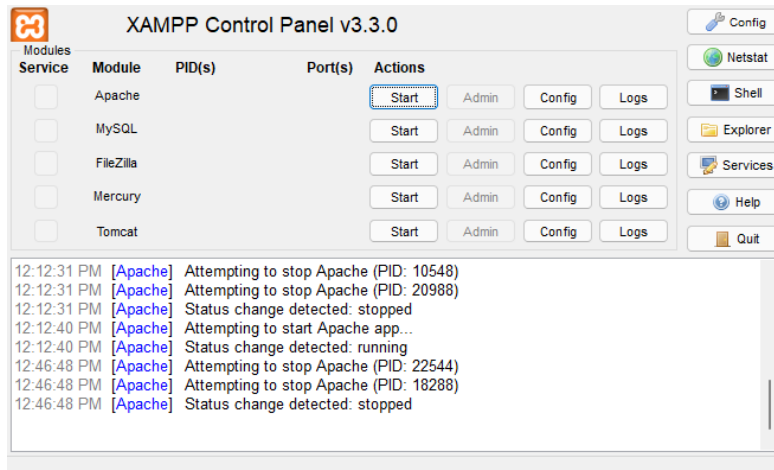
15. Ensure your VM allows incoming connections on port 80.

16. On your PC, test your connection to the web server by navigating to the media server on a web browser.

Using the above example IP, you would try to navigate to `http://3.238.18.42:80`. If it displays something like this:

```
[
    "rock",
    "jazz",
    "bar",
    "endgame",
    "clockwork",
    "emagged",
    "shuttle",
    "muzak",
    "trance",
    "beach",
    "delta",
    "lobby-main",
    "lobby-snow",
    "nukesquad",
    "malfdelta"
]
```

Then the media server is now running. However, because the git repo doesn't have any of the actual data for /vg/'s playlists, it won't actually be able to serve any music or playlists. We'll clean out the existing playlists.json file along with some other miscellaneous files:

1. On your VM, navigate to the XAMPP server root again (`C:/xampp`).
2. Use a text editor to open `playlists.json`. Delete all of the playlist information between the curly braces *except* for the playlist information regarding "emagged". (The "emagged" playlist is hard-coded in /vg/ station). Your `playlists.json` should just look something like this:

```
{
  "emagged": {
```

```
        "obfuscate": "true"
    }
}
```
3. Clean up code/modules/media/jukebox.dm (there should be a commit/PR on github for this as of 7/20/21) and leftover records in the code..

Configuring /vg/station to use the media server
1. On your VM, open the /vg/station folder.
2. Open `config/config.txt` with a text editor. Set MEDIA_BASE_URL to the ROOT_URL you used above. Using the example IP, the following lines should be set and uncommented (i.e. remove the beginning hash):
`MEDIA_BASE_URL http://3.238.18.42:80`
3. If you set API_KEY above when configuring the media server, set it here too. For example, if API_KEY is "iwtcird", set this to:
`MEDIA_SECRET_KEY iwtcird`
4. Note that this file should not be pushed to a git repo if you want to keep API_KEY a secret.

mare

Editing playlists/music
Changes made to playlists and music on the media server will at minimum **not** be reflected on the server until new jukeboxes are spawned.

Method 1: Using the media converter tool
/vg/ coders "provide" a media converter tool that converts .ogg, .mp3, and .m4a files (NOTE: .wav files are not supported because idk) into smaller .mp3 files, collects music metadata, and bundles them in the correct filesystem structure for the media server to use. All of the following steps should be performed on your VM.

Part 1. Setup Python/sox/ffmpeg/libmp3lame (only needs to be done once)
On your VM:
0. Install 7zip from https://www.7-zip.org/ if it is not already on your VM.
1. Install python from https://www.python.org/downloads/ if it is not already on your VM.
2. Open a run prompt with `Win+R`.
3. Open PowerShell by typing `powershell` and press `Return`.
4. Download necessary packages in PowerShell using pip, the python package manager:
   `pip install pyyaml mutagen pybuildtools`
5. Download and install the **32-bit** sox for Windows:
https://sourceforge.net/projects/sox/files/latest/download
6. Close PowerShell.

7. Open an elevated command prompt by pressing `Win+X` and selecting "Command Prompt (Admin)" on the menu that appears.

8. Type `rundll32.exe sysdm.cpl,EditEnvironmentVariables` and press `Return`. OR:

On Windows 10, open up search and type "environment". Select "Edit the system environment variables". On the dialog that pops up, click "Environment Variables…"
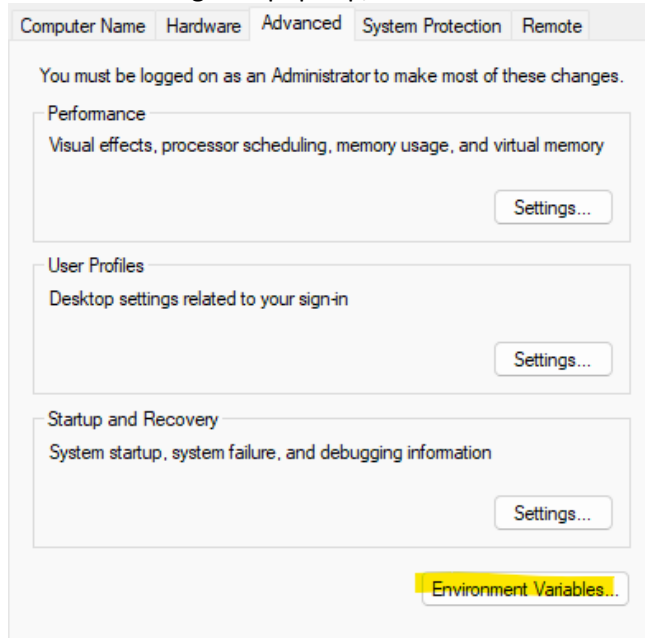
9. Under "System Variables" (not "User Variables"), scroll to find and select the entry that says "Path" and click "Edit…"

10. On the dialog that pops up, click "New"

11. Add the sox installation folder to your Path (add the corresponding path for your version of sox:)

```
C:\Program Files (x86)\sox-…
```

12. Click "OK". Do not close the environment variable window as we will use it later.

13. Test that sox has been added to your path. Open another PowerShell prompt by following steps 2 and 3 again.

14. Run:

```
sox
```

If an error message does not appear, sox is properly installed.

15. Download https://archive.org/download/libmp3lame-0.dll (smutty.horse mirror: https://u.smutty.horse/mceigmrybpb.dll) and drop it into the sox installation folder. This is a library used for mp3 encoding:

```
C:\Program Files (x86)\sox-…\libmp3lame-0.dll
```

16. Close PowerShell.

17. Download an ffmpeg release for Windows: https://www.gyan.dev/ffmpeg/builds/

18. Use 7zip to extract the ffmpeg release (win64) to a location of your choice.

19. Repeat steps 10-12, but instead of the sox installation folder, add the path to your ffmpeg bin folder:

```
C:\...\ffmpeg-2021-07-18-git-694545b6d5-full_build\bin
```

20. Repeat steps 2 and 3 and run:

```
ffmpeg
```

If an error message does not appear, ffmpeg is properly installed.

Part 2. Setting up the media converter (only needs to be done once)
1. Using git or zip download, clone the SS13 Media Converter folder (`ss13-media-converter`) somewhere on your VM.

> git clone https://github.com/vultraz168/ss13-media-converter.git
> OR download:
> https://github.com/vultraz168/ss13-media-converter/archive/refs/heads/master.zip

2. In the `ss13-media-converter` folder, create an empty folder called `files-upl`

Part 3. Tagging your input files
For an example input file, we're going to use the song "It's Gonna Work" from the sixth season of My Little Pony: Friendship is Magic. Download here:
https://u.smutty.horse/mceinlwpbkr.ogg
1. The media server and converter use metadata built into the music files to function properly. This metadata can be provided using a tagging software, such as Mp3tag, which can be downloaded from this link: https://www.mp3tag.de/dodownload.html
2. Drag and drop the audio file into Mp3tag.
3. We're going to assign some metadata to this file. On the left side, enter in a title. Then click the save icon on the top left corner or press `Ctrl+S` to save the tags. The file is now ready for conversion.

Part 4. Using the media converter
1. We're going to put this track into a new playlist called "pony". Open up `config.yml` in `ss13-media-converter` and add a new line after `playlists` reading:
> pony: {}

(This does not have to be done for adding a track to an existing playlist).
2. To convert the .ogg file, we begin by creating a folder called "pony" in the "source" folder if it does not already exist. This folder will hold all the input files for the "pony" playlist:
> ss13-media-converter/source/pony

3. Move the .ogg file into the folder of `ss13-media-converter`
> ss13-media-converter/source/pony/mceinlwpbkr.ogg

4. Open up PowerShell and `cd` to the `ss13-media-converter` directory.
5. Run:
> python convert.py

If all is well, the logging will not spit out an error at the end. The previously empty `files-upl` folder should contain two folders: `B` and `fileData.json`.

Part 5. Pushing songs to the media server
Simply copy the contents of `files-upl` into the `files` directory on the XAMPP server root.
`C:/xampp/files` should then have the same contents as `files-upl`:
`C:/xampp/files/B`
`C:/xampp/files/fileData.json`

Note: This will overwrite an existing `fileData.json` file. If you need to merge in converted files from a different source (say, generated by somebody using a media converter on another computer) onto an existing `fileData.json`, you will need to manually edit the fileData.json.

Part 6. Adding new playlists to the media server and BYOND

In addition to `config.yml`, playlist names and metadata are stored inside the **`playlists.json`** file mentioned above (*Configuring and running the media server on Windows*). The playlist json does not store actual track information. An example of the `playlists.json` data is shown below:

```
{
    "rock": {},
    "emagged": {
        "obfuscate": "true"
    }
}
```

This `playlists.json` file tells the media server that there is one playlist called `rock` and one playlist called `emagged`. The `emagged` playlist has the `obfuscate` variable set, meaning that the media server will hide the artist and album names of tracks in the `emagged` playlist.

In addition to `playlists.json`, playlist information is additionally stored in BYOND code under `code/modules/media/jukebox.dm`, specifically under the global variable **`global_playlists`** (to tell the jukebox which playlists to fetch from the media server) and under the **jukebox** classes (to tell the game which playlists are permitted to play on the jukebox). ***All four locations (config.yml, playlists.json, global_playlists, jukebox classes must be updated in order for playlists to work.***

Note: I don't understand why the system was designed like this but that's just how it is.

Part 7. Meta
The likely meta for uploading music is as follows:
- The host maintains the music library on the server as well as in the `source` folder for the converter. This music library should be backed up somewhere in case a migration is needed.
- People who wish to upload music should tag their music files with mp3tag as described above, providing track titles and optional artists/albums/track numbers. Then they put their files into a .zip and send that to the host.
- The host unzips the files into the appropriate playlist folder in `source` and runs the converter.
- If a new playlist is needed, either the host makes the appropriate modifications in `code/modules/media/jukebox.dm` or the person who wants to create the new playlist should make a PR with the necessary changes. The host will also have to update `config.yml` in the media converter and `playlists.json` on the server side.

Procedure:
Note that in the above "adding tracks" example, we placed "It's Gonna Work" in the `pony` playlist. However, the playlist itself is not yet on the media server.
1. First, we must add the playlist name to `playlists.json`:

```
{
    "pony": {},
    "emagged": {
        "obfuscate": "true"
    }
}
```

2. To update the media server cache for a particular playlist, (using the above example IP), open a web browser and navigate to
http://3.238.18.42:80/index.php?playlist=pony&reset_cache

3. Because this playlist is not in the /vg/ codebase, we must also add it to /vg/station.
4. As mentioned above, you need to modify at least two locations in `code/modules/media/jukebox.dm`: **`global_playlists`** and the **`jukebox`** classes. For each one, there is a list of playlists which you need to add the playlist name to:
Before:
```
for(var/playlist_id in list(…))
```
After:
```
for(var/playlist_id in list("pony",…))
```
This ensures the playlists are globally accessible by the game.

Before:
```
Playlists=list(
      …
)
```
After:
```
Playlists=list(
      …,
      "pony" = "The Pony Playlist"
)
```
5. If you have modified the code, you must recompile the /vg/station server.

mares

--UNTESTED-- Method 2: Manually
Just in case the media converter doesn't work, there's always the fun way…

Adding tracks
Background information
Music files are stored in the `files` directory in the XAMPP server root (*Configuring and running the media server on Windows*). The media server organizes files by their uppercase md5 hashes in a three-tiered folder structure (convert.py, line 164;). For instance, suppose the md5 hash of an mp3 file is `99B86B9139CBE2DE2C98320EC258540A`

The file is then stored in:
`files/9/9/B86B9139CBE2DE2C98320EC258540A.mp3`

Music metadata is located in a JSON file called `fileData.json`, placed in the `files` folder mentioned above. This file is not included in the git repo by default; you will have to create it if it is not already there. Here is an example of the contents of `fileData.json:`

```
// Comments are written in green.

["99B86B9139CBE2DE2C98320EC258540A ": {
    "title": "You Have To Give Up"
    "album": "You Have To Realize You Will Die" // optional
    "artist": "Tyler Discord" // optional
    "tracknumber": "20" // optional
    "playtime_seconds": "111" // floating point, decimals allowed.
    // length sent to BYOND by index.php is deciseconds.
    "playlists": ["jazz"] // this is where playlists are specified
    "orig_filename": "you_are_useless.mp3" // optional
},]
```

Essentially, this is an array of objects that represent track data. Each track is indexed by its md5 hash (99B86B9139CBE2DE2C98320EC258540A). **Note that the data that associates tracks with playlists is attached to tracks, not playlists.**

Procedure

1. Tracks should first be transcoded to .mp3 format, as all of the media server code is written to use mp3 files. You can use ffmpeg, Audacity with LAME, other transcoding tools, or an online conversion utility to do this.

2. Obtain the md5 hash of the track.

Method 1. There are [websites that allow you to upload a file](#) and obtain its hash if you're really afraid of terminals, although you should obviously take into account that you are uploading a file to a random website.

Method 2. md5 hashes can also be performed natively using the `certutil` command.

    1. Open a run prompt with `Win+R`.

    2. Open PowerShell by typing `powershell` and pressing Return.

    3. `cd` to the directory with your track. For example:

        `cd D:\MLP_Samples\Songs`

    4. Run the `Get-FileHash` command on the file, specifying the md5 algorithm. For example, in `D:\MLP_Samples\Songs` I have a file called `Giggle At The Ghosties.mp3`. I would then type:

        `Get-FileHash 'Giggle At The Ghosties.mp3' -Algorithm md5`

Note: If you forget the 'md5' at the end you will get a different hash.

    You should get something that looks like the output below. The md5 hash is highlighted:

```
Algorithm       Hash                                                Path
---------       ----                                                ----
MD5             CB326FB10737B02FCD1CF9B0788E571D                    ...
```

3. Now that we have the md5 hash of the file, we can add the file to the media server.

4. If it does not already exist, create a directory under `files` for the first character of the md5 hash. For the file above, this would be:
`C:\xampp\files\C`

5. If it does not already exist, create a directory under `files` for the second character of the md5 hash. For the file above, this would be:
`C:\xampp\files\C\B`

6. Copy and paste the mp3 file into this directory and rename it with the remaining characters of the md5 hash and the mp3 extension:
`C:\xampp\files\C\B\326FB10737B02FCD1CF9B0788E571D.mp3`

7. Add metadata for the file to `fileData.json`. If it does not already exist, create it here:
`C:\xampp\files\fileData.json`

The format for `fileData.json` is shown here. Note that the md5 hash is highlighted where it needs to appear. Also note that the track is placed in the `pony` playlist:

```
[

"md5 hash 1": { blah blah blah },

"md5 hash 2": { blah blah blah },

"326FB10737B02FCD1CF9B0788E571D": {
      "title": "Giggle At The Ghosties"
      "album": "" // optional
      "artist": "Pinkie Pie" // optional
      "tracknumber": "1" // optional
      "playtime_seconds": "70.53" // floating point, decimals allowed.
      // length sent to BYOND by index.php is deciseconds.
      "playlists": [ "pony", "emagged" ] // this is where playlists are
specified.
      "orig_filename": "Giggle At The Ghosties.mp3" // optional
},

"md5 hash 3": { blah blah blah },

]
```

You have now added the track to the media server. To get the server to update the cache for a particular playlist, (using the above example IP), open a web browser and navigate to
http://3.238.18.42:80/index.php?playlist=pony&reset_cache

Maintenance
1. Any time the VM is restarted, you should remember to turn on the media server.
2. To update the server after modifying a playlist, (using the above example IP), open a web browser and navigate to http://3.238.18.42:80/index.php?playlist=pony&reset_cache.
Changes will not be reflected until a new jukebox is created (SS13 round restart, SS13 server restart, or admin spawn).

Changelog
7/20/21: Fixed reset_cache link, forked media server repo to fix bugs, added activex section, added XAMPP link, added info about cleaning records, added meta section, fixed bugs with both media server repo and media converter.