

Program Launch Arguments

launch arguments

You can launch the program with several different arguments to alter core behaviour. If you are not familiar with this, you are essentially putting additional text after the launch command that runs the program. You can run this straight from a terminal console (usually good to test with), or you can bundle it into an easy shortcut that you only have to double-click. An example of a launch command with arguments:

```
“ C:\Hydrus Network\client.exe -d="E:\hydrus db" --no_db_temp_files
```

You can also add --help to your program path, like this:

```
“ client.py --help
```

```
server.exe --help
```

```
./server --help
```

Which gives you a full listing of all below arguments, however this will not work with the built client executables, which are bundled as a non-console programs and will not give you text results to any console they are launched from. As client.exe is the most commonly run version of the program, here is the list, with some more help about each command:

- **-d DB_DIR, --db_dir DB_DIR**

Lets you customise where hydrus should use for its base database directory. This is install_dir/db by default, but many advanced deployments will move this around, as described [here](#). When an argument takes a complicated value like a path that could itself include whitespace, you should wrap it in quote marks, like this:

```
“ -d="E:\my hydrus\hydrus db"
```

- **--temp_dir TEMP_DIR**

This tells all aspects of the client, including the SQLite database, to use a different path for temp operations. This would be by default your system temp path, such as:

“ C:\Users\You\AppData\Local\Temp

But you can also check it in *help->about*. A handful of database operations (PTR tag processing, vacuums) require a lot of free space, so if your system drive is very full, or you have unusual ramdisk-based temp storage limits, you may want to relocate to another location or drive.

- **--db_journal_mode {WAL,TRUNCATE,PERSIST,MEMORY}**

Change the *journal* mode of the SQLite database. The default is WAL, which works great for SSD drives, but if you have a very old or slow drive, a different mode *may* work better.

Full docs are [here](#).

Briefly:

- WAL - Clever write flushing that takes advantage of new drive synchronisation tools to maintain integrity and reduce total writes.
- TRUNCATE - Compatibility mode. Use this if your drive cannot launch WAL.
- PERSIST - This is newly added to hydrus. The ideal is that if you have a high latency HDD drive and want sync with the PTR, this will work more efficiently than WAL journals, which will be regularly wiped and recreated and be fraggy. Unfortunately, with hydrus's multiple database file system, SQLite ultimately treats this as DELETE, which in our situation is basically the same as TRUNCATE, so does not increase performance. Hopefully this will change in future.
- MEMORY - Danger mode. Extremely fast, but you had better guarantee a lot of free ram.

- **--db_cache_size DB_CACHE_SIZE**

Change the size of the cache SQLite will use for each db file, in MB. By default this is 200, for 200MB, which for the four main client db files could mean 800MB peak use if you run a very heavy client and perform a long period of PTR sync. This does not matter so much (nor should it be fully used) if you have a smaller client.

- **--db_synchronous_override {0,1,2,3}**

Change the rules governing how SQLite writes committed changes to your disk. Full docs [here](#). The hydrus default is 1 with WAL, 2 otherwise.

- **--no_db_temp_files**

When SQLite performs very large queries, it may spool temporary table results to disk. These go in your temp directory. If your temp dir is slow but you have a *ton* of memory, set this to never spool to disk, as [here](#).

- **--boot_debug**

Prints additional debug information to the log during the bootup phase of the application.

- **--no_daemons**

Launch the program without some background workers. This is an old debug command and does not do much any more.

The server supports the same arguments. It also takes a *positional* argument of 'start' (start the server, the default), 'stop' (stop any existing server), or 'restart' (do a stop, then a start), which should go before any of the above arguments.

Revision #1

Created 12 March 2021 16:44:59 by CuddleBear

Updated 12 March 2021 18:24:07 by CuddleBear