**POE Map Watch Script**

**Features:**

* Ctrl+C on maps to insert map data for each map
  + Open command prompt
    - Show map selected/copied
    - ===
    - Map Currently Running (name or none) \*clear map?
    - Button - Change to Map Above
    - Button - Clear Map / Map Cleared
    - ===
    - Add Map to Database
    - Start New Map Database???
* Parse map data from clip board
  + Place all data into file/database
  + Unique Key: Local time
  + Map Dropped From/In
    - Can be set to None
    - Linked to another map table
  + Map Name
  + Map Tier
  + Map Level (obtained from Map Tier now)
  + Map Rarity (normal/magic/rare/unique)
* Map Running Table
  + Unique Key: Local time
  + Map Name
  + Map Tier
  + Map Level (obtained from Map Tier now)
  + Map Rarity (normal/magic/rare/unique)
  + Map IIQ
  + Map IIR
  + Map Pack Size
  + Map Quality
  + Map Mods...
* Start and end times for comparison
  + Hotkeys for starting and ending runs
* Settings:
  + Map check interval time
  + Time-zone
  + Load last used database file on startup
* Generate Statistics File
  + **Average map drops per map ran**
    - SELECT Found\_In FROM Maps\_Dropped WHERE Found\_In > 0
    - SELECT Time\_Stamp\_ID, IQ FROM Maps\_Ran
    - Loop through list pairing Found\_In with number of times it appears
    - [{IQ:drops},..]
  + **Each map ran and its maps dropped (done)**
    - All maps ran in a combo box
    - SELECT \* FROM Maps\_Ran
    - Select a map and see all its info with mods
    - Under each map selected show its maps dropped by Tier, Name, Rarity
    - SELECT Tier, Name, Rarity FROM Maps\_Dropped
    - Easily put into a table, but could look better with some images and CSS
      * Map graphic and arrow pointing to each map drop
      * Each map drop will me a smaller simpler map graphic
    - [ {'Map Ran':{\*}, 'Maps Dropped':{ {Tier, Name, Rarity},...} },...]
  + Average map drops per hour(s)
    - Note:
    - Time\_Stamp\_ID of every map drop
    - Convert first and last Time\_Stamps\_ID too hours and L - F = hours\_recorded
    - Math: (*this has to be in Session option*)
    - TBA
  + **Number of +0 +1 +2 Map Tier drops in maps ran (done)**
    - **Best Map returns (use Tier and IQ of map ran)**
    - SELECT Time\_Stamp\_ID IN Maps\_Dropped WHERE Found\_In\_ID > 0
    - Loop through list pairing Found\_In\_ID with each map drop Time\_Stamp\_ID
      * [{'Map\_Tier': 7, 'Tier\_Drops': [6,4,9,7,8], ~~'Drops': [12,13,14]}~~,...]
    - Table X: -3 or less, -2, -1, 0, +1, +2
    - Table Y: Map Tiers 1-15 (stack multiple tiers?)
    - Bar Chart that uses multiple bars per tier (if needed)
  + **Amount of each map tier dropped (done)**
    - **All Map Drops**
    - SELECT Map\_Tier IN Maps\_Dropped
      * [1,2,3,4,4,5,6,7,4,3,4,2,1,3,2,1...]
      * Loop and add one to each index (tier)
      * Might have to create 16 index array beforehand, else use object
      * {0:0, 1:34, 2:30, 3:33, 4:28, 5:27,...]
    - Line Chart
  + **Average map drop in map IQ range (50-59, 60-69, etc) (Done)**
    - Option: exclude unidentified maps (only saved at +30% IQ, and not its actual IQ)
    - SELECT Time\_Stamp\_ID, IQ IN Maps\_Ran WHERE Mod1 NOT 'Unidentified'
    - SELECT Time\_Stamp\_ID IN Maps\_Dropped WHERE Found\_In\_ID > 0
    - Loop through each query
    - Return [ [map ran time\_stamp, IQ, maps dropped], ... ]
    - If q[0][1] in range ?-? then add map drop count to that array
      * Ex. 60-69% array will have [3 maps ran, 11 maps dropped]
      * This can all be kept in one bigger object {'60-69% IQ': [^]}
      * Then do average math calculations while looping through this to display info in charts/tables/etc
  + Average map drop per map tier/rarity
    - This is the same as an IQ range
  + Best to worst maps ran in regard to map drops
    - I think this will be a variant of +1+2 Map Tier drops
  + Order of map drops by:
    - Map ran IQ
    - Map Tier
    - Amount (Map Name)
* Session Option
  + A spinner option box will show how long a session will be (hours)
  + Timer will need to be shown in corner. Always on top and very small.
  + Session column name will record a time stamp at start of session
  + Each map drop will have this starting time stamp

Map\_Time\_ID

Map\_Found\_In\_ID

Removes the map currently being run

Preferences

Choice

Averaged over

Behavior

General

Language

Corrupted

Breakeven

Avg Returns per Map

Map Drops by Tier

Map Drops by IQ

All Map Data

Description:

Map Watch is an application that records map drops in the game Path of Exile. Once enough data is collected it can show different statistics on the map drops recorded.

Use:

You can run the Python script in version 3.4 or download the full release that includes a Windows exe that any Windows PC should be able to run. While in game the application will run in the background and it will popup once a player finds and copies a map's info to the clipboard. To do this, simply highlight a map in your inventory and press Ctrl+C. After that the application is pretty obvious I hope.

Important Note:

It is highly recommended you play Path of Exile in a Windowed Mode while using this application. Playing in Full Screen will cause problems that may lead to your death in game (i.e. the game might minimize). Don't say I didn't warn you.

Versions:

0.1 Initial Release

Licensing:

Please see the file called LICENSE.

Contacts:

Email: Jonathan.D.Hatten@gmail.com

Path of Exile IGN: Grahf\_Azura

Map Watch

Version 0.1

Created by

Jonathan.D.Hatten@gmail.com

IGN: Grahf\_Azura

Apache HTTP Server

What is it?

-----------

The Apache HTTP Server is a powerful and flexible HTTP/1.1 compliant

web server. Originally designed as a replacement for the NCSA HTTP

Server, it has grown to be the most popular web server on the

Internet. As a project of the Apache Software Foundation, the

developers aim to collaboratively develop and maintain a robust,

commercial-grade, standards-based server with freely available

source code.

The Latest Version

------------------

Details of the latest version can be found on the Apache HTTP

server project page under http://httpd.apache.org/.

Documentation

-------------

The documentation available as of the date of this release is

included in HTML format in the docs/manual/ directory. The most

up-to-date documentation can be found at

http://httpd.apache.org/docs/trunk/.

Installation

------------

Please see the file called INSTALL. Platform specific notes can be

found in README.platforms.

Licensing

---------

Please see the file called LICENSE.

Contacts

--------

o If you want to be informed about new code releases, bug fixes,

security fixes, general news and information about the Apache server

subscribe to the apache-announce mailing list as described under

<http://httpd.apache.org/lists.html#http-announce>

o If you want freely available support for running Apache please see the

resources at <http://httpd.apache.org/support.html>

o If you have a concrete bug report for Apache please see the instructions

for bug reporting at <http://httpd.apache.org/bug\_report.html>

o If you want to participate in actively developing Apache please

subscribe to the `dev@httpd.apache.org' mailing list as described at

<http://httpd.apache.org/lists.html#http-dev>