Overview

This program is a companion program to JS8Call Monitor. It is designed to listen for messages and spot reports from JS8Call Monitor, then send those to a GeoChron digital display via the GeoChron servers.

Prerequisites

Python 2.7 or higher
GeoChron Digital 4K, Atlas 2 or higher (https://www.geochron.com/4k/)
GeoChron API account
Internet Connection

Upgrading From a Previous Version

To upgrade your previous installation, start by locating the folder on your computer where you installed GeoServer.

Rename the config file to config.old.

Unzip the contents of the geoserver.zip file to a suitable place on your computer.

Copy the new GeoServer files into your installation folder, replacing the old files.

Using your favorite text editor, edit the config file for your setup. You can reference the config.old file for your previous settings, and refer to the section below on any new settings.

New Installation

Unzip the contents of the geoserver.zip file to a suitable place on your computer.

Using your favorite text editor, edit the config file for your setup.

- The **GeoChron** section contains various setting that control the operation of GeoServer.

  base_url contains the base https url to the GeoChron API server. It should not be necessary to change this field.

  api-key is for the unique API key associated with your GeoChron display. See the section on setting up your API account below for more information.

  contacts_layer is for the numeric number of the contacts layer. See the section on setting up your API account below for more information.

contacts_component is for the name of the CSV component for contacts. See the section on setting up your API count below for more information.

messages_layer is for the numeric number of the messages layer. See the section on setting up your API account below for more information.

messages_component is for the name of the text legend component for messages. See the section on setting up your API account below for more information.

messages_position sets the position of the message window on the GeoChron display. The default position is Top.

messages_color sets the text color of messages on the the GeoChron display. This value should be set in color hex notation I.E. #ff00ff

messages_font_size sets the font size of messages. The default value is 60.

grid_nocall sets the contact (spot) text displayed when no call sign is received.

grid_nocolor sets the color of the contact (spot) when so color data is received.

- The **GeoServer** section contains the details of how JS8Call Monitor will connect to GeoServer.

host should be set to the IP address of the host where you are running GeoServer. If you are running GeoServer on the same host as JS8Call Monitor you can use 127.0.0.1.

port should be set to the port GeoServer will be listening on for messages.

- The **Auth** section contains the details of how JS8Call Monitor will authenticate with GeoServer.

enabled flag if set to true, requires JS8Call Monitor authenticate all requests by passing a security token as part of the request. You must be running version 0.20 or higher of JS8Call Monitor for this feature to work.

token is used to specify the security token (password).

- The **DEBUG** section has various flags for debugging the script. See the section on debugging for an explanation of these parameters.

Save your changes.

Configuring JS8Call Monitor to Use GeoServer

On the computer where you are running JS8Call Monitor, navigate to the folder where you installed JS8Call Monitor.

Using your favorite text editor, edit the config file for your setup.

[GEOSERVER]
enabled_msg = true
enabled_spot = true
host = 127.0.0.1
port = 23000
token = fixme

- The **GEOSERVER** section contains the settings for how JS8Call Monitor connects to GeoServer. The settings in the GEOSERVER section of the JS8Call Monitor config file should match those set in the GeoServer and Auth sections of the GeoServer config file.

  Change **enabled_msg** to true, to enable JS8Call Monitor sending message updates to GeoServer.

  Change **enabled_spot** to true, to enable JS8Call Monitor sending spot (contact) reports to GeoServer.

Save your changes.

Restart JS8Call Monitor if it is running.

Running the Program

Before running the GeoServer script for the first time you will need to setup and configure the following;

- The GeoChron Digital Display (see below).
- The GeoChron API account (see below).

Run the script from Python. This can be done by any of the following methods;

a) from the OS command line "Python geoserver.py"
b) loading it in IDLE (Python GUI), then running the script.
c) by right clicking in Windows and select Open with Python
d) by right clicking in Windows and select Open in IDLE, then running the script

When you first start the script, if it complains about missing modules, you can get the missing ones by running "python -m pip install <module>" from an OS command prompt.

As JS8Call Monitor messages are received and processed, the GeoChron display should be automatically updated with the contacts (spots) and messages.

Setting Up The GeoChron API Account.

Before you can send data to the GeoChron digital display, you will need to set up an API account. At the time this document was prepared there was no fee for this service.

Before you start the API registration, setup and register your GeoChron digital display. You will need to have a working display to complete the API account registration process.

Open a web browser and navigate to https://interface.geochron.com.

Click on the Register link in the menu at the top of the page.

Fill out the page and click Next.

The next page will prompt you for details about your GeoChron display.

Fill out the page and click Next.

The last page will prompt you for a PIN. It mentions that PIN should show up on the System tab on your display. There is currently a bug, and the pin may actually show up on the Live tab.

Enter the pin displayed on the GeoChron display, without hyphens and click next.

That should complete the API account registration process.

Configuring The GeoChron API Account

If you aren't already, log in to the API account by clicking the Login link in the menu at the top of the page.

Enter the email address and password you used when you created the account.

Click on the Your Layers link in the Top Menu.

You should now see under Your Custom Layers, one layer named with the email you used for registration, and a big button that says New Layer. **What ever you do, do not delete the layer named with the email address.** If you do the New Layer button will disappear.

Click the New Layer button.

You should now have two layers and a New Layer button Under your Custom Layers.

Start with the layer named with the email address. Click on it. It should open up in a full browser window.

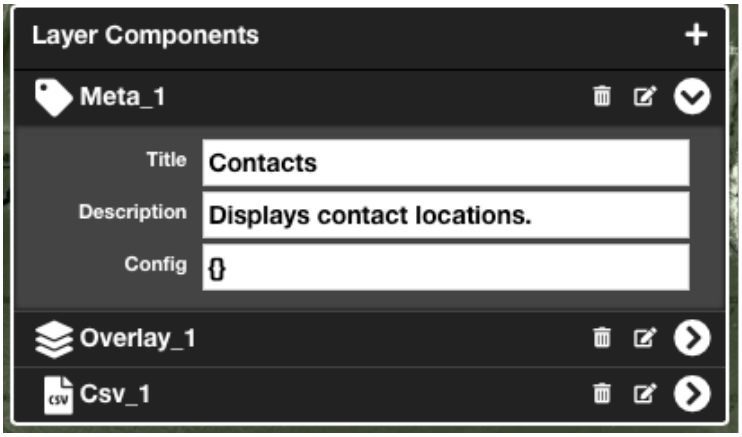
Click the Edit Layer button in the top menu.

Click on the pencil icon next to the name and change it to Contacts, or something suitable. The name of the layer is not terribly important. This layer will be used to send contact, or spot reports to your GeoChron display.

Add the following components to this layer;

One Meta component.
One Overlay component.
One CSV component.

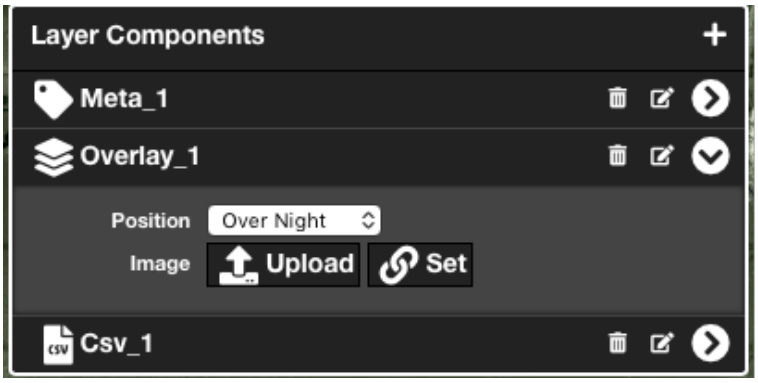
Open the Meta component, by clicking on the word Meta. It should open up and look like below.



Fill in the Title and Description fields. What you put here determines how this option will show up in the Live tab of your GeoChron display configuration. You can put whatever you want here.

Save it by clicking on the down arrow button.

Open the Overlay component. Make sure the Position field is set to Over Night like below.



Save it by clicking the down arrow.

You do not need to make any further changes to this overlay.

Note the exact name given to the CSV component I.E. Csv_1 and the overlay number at the end of the URL shown in the browser I.E. 123 if the url is https://interface.geochron.com/layer/123.

Now save the changes to the layer by clicking the save changes button at the top of the screen.

Now let's configure the layer for the messages.

Click on the Your Layer link in the top menu to return to the Layers page.

Open the new layer you previously created by clicking on it.

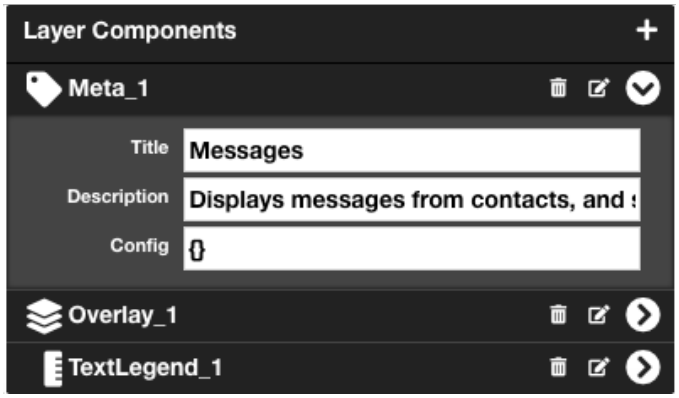
Click Edit Layer in the top menu.

Click on the pencil icon and change the name of the layer to Messages or something appropriate. The name of the layer is not terribly important. This layer will be used to send messages to your GeoChron display.

Add the following components to this layer;

One Meta component.
One Overlay component.
One Text Legend component.

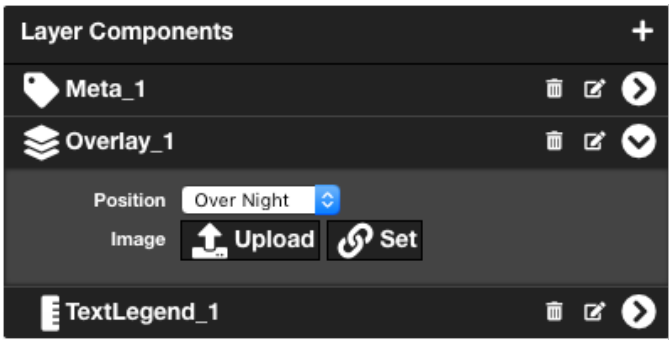
Open the Meta component, by clicking on the word Meta. It should open up and look like below.



Fill in the Title and Description fields. What you put here determines how this option will show up in the Live tab of your GeoChron display configuration. You can put whatever you want here.

Save it by clicking on the down arrow button.

Open the Overlay component. Make sure the Position field is set to Over Night like below.



Save it by clicking the down arrow.

You do not need to make any further changes to this overlay.

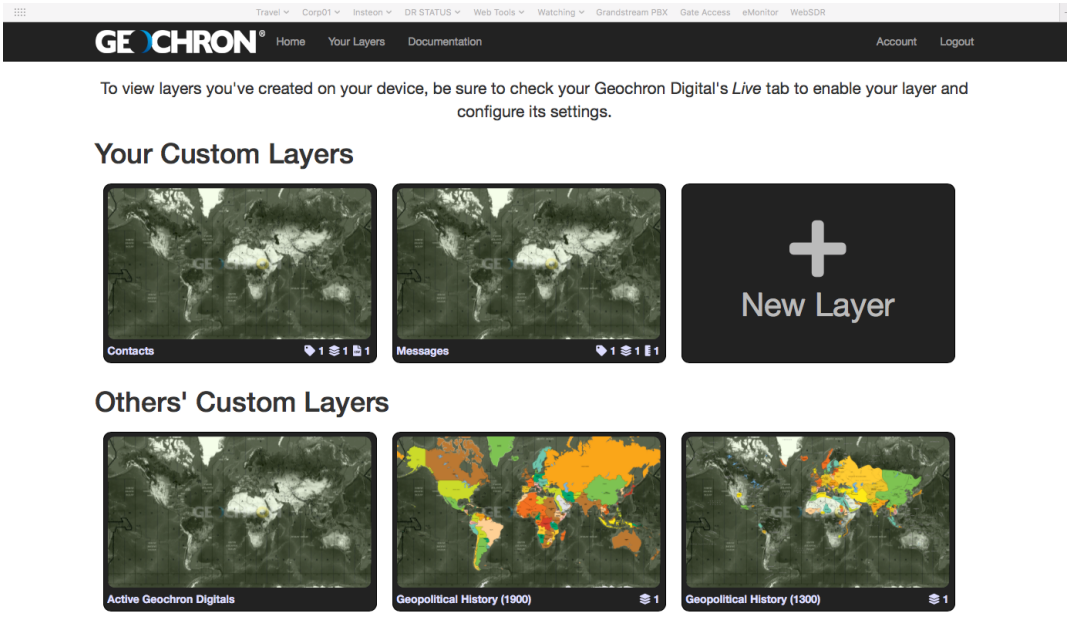
Note: There is a bug with the Text Legend component. Do not attempt to open it, or it will crash the page and you will have to start over.

Note the exact name given to the Text Legend component I.E. TextLegend_1 and the overlay number at the end of the URL shown in the browser I.E. 124 if the url is https://interface.geochron.com/layer/124.

Now save the changes to the layer by clicking the save changes button at the top of the screen.

Now click the Your Layers link in the top menu.

Your screen should now look like below.



Updating the GeoServer Configuration

Open the GeoServer config file using your favorite text editor.

- In the GeoChron section make the following edits.

api_key - Fill in this field with your api key. It can be found by clicking the account link in the top menu of the GeoChron API page.

contacts_layer - Fill in this field with the number you noted when creating the contacts layer I.E. 123.

contacts_component - Fill in this field with the exact name you noted for the CSV component when you created the contacts layer. I.E. Csv_1

messages_layer - Fill in this field with the number you noted when creating the messages layer I.E. 124.

messages_component - Fill in this field with the exact name you noted for the Text Legend component when you created the messages layer. I.E. TextLegend_1

Save the GeoServer configuration.

Start GeoServer (see the Running the Program section above.)

Configuring Your GeoChron Display

The last step is to configure your GeoChron display.

Using your remote open the menu of your GeoChron digital display.

Navigate to the Live tab.

You should see the API layers at the bottom of the list.

Enable each API layer and set the options to your preferences.

Close the menu.

Congratulations! GeoServer and your GeChron display are now configured.

How It Works

JS8Call Monitor version 0.8 and above supports integration with GeoServer. Currently JS8Call Monitor supports sending grid based spot reports to mapping applications, so they can be visually mapped using County Mapper and/or GridTracker. GeoServer now extends this capability to GeoChron digital displays.

GeoServer also adds a feature unique to GeoChron displays. When a JS8Call INFO message is received by JS8Call Monitor, it can send an abbreviated version of the INFO message to GeoServer, that in turn will send the message to your GeoChron digital display.

The GeoServer listens for JSON formatted messages on the port configured in the GeoServer config file.

When it receives a message, it formats it and then calls the appropriate GeoChron API, which then queues the update for your GeoChron display.

The GeoChron API's are currently cloud based, so an Internet connection is required for this setup to work.

The GeoChron digital display periodically polls the GeoChron servers for updates. When it finds one or more, it then updates the display. Note: It can take up to 10 minutes for updates to appear on your display.

While this may not be the most efficient method, it is the only method currently offered by GeoChron. But the cool factor is a 10.

Clearing Contacts (Spots) and Messages From Your GeoChron Display

The current design of JS8Call Monitor, transmits the grid squares to be displayed to the configured mapping applications. Over time these displayed grid squares accumulate, and the mapping applications needs to be cleared manually. GeoServer is no different in this regard.

Included in the distribution are two python files spots_clear.py and messages_clear.py. Open each file in your favorite text editor, and edit the host, port, and token information to match that of your GeoServer configuration, then save the files.

To clear the spots you simply run the spots_clear.py file. This can be done from the command line I.E. "Python spots_clear.py".

To clear the messages you simply run the messages_clear.py file. This can be done from the command line I.E. "Python messages_clear.py"

A word of caution. Clear the messages and spots from GeoSever before shutting down GeoServer. If you shut down GeoServer first you will need to manually clear the "Points" that were created on the Contacts layer by editing the layer on the GeoChron API web site. Messages however can be cleared after restarting GeoServer by running the messages_clear.py file. Unfortunately, this is a limitation of the current GeoChron API.

Pro tip: Add the spots_clear.py and messages_clear.py to a cron job (Linux or Pi) or scheduled task (Windows). I have my system set to run each script at midnight, to clear the messages and spots (contacts) from the previous day.

Other Uses For GeoServer

While GeoServer was primarily designed to display messages from JS8Call Monitor, it can run stand alone, and serve other purposes as well. An example client is furnished in the distribution. Commands are sent as JSON formatted messages over a UDP connection. The following are the commands supported by GeoServer.

GeoServer Commands

MESSAGE.CLEAR - Clears all messages from the display.

```
data = {
        'type': 'MESSAGE.CLEAR',
        'params': {'AUTH': 'fixme'}
}
```

MESSAGE.TEST - Send a test message to the display.

```
data = {
        'type': 'MESSAGE.TEST',
        'params': {'AUTH': 'fixme'}
}
```

MESSAGE.SEND - Send text message to the display. There is a 40 character limit.

```
data = {
        'type': 'MESSAGE.SEND',
        'params': {'AUTH': 'fixme', 'MESSAGE': 'Hello World!'}
}
```

SPOT.CLEAR - Clears contacts (spots) from the display.

```
data = {
        'type': 'SPOT.CLEAR',
        'params': {'AUTH': 'fixme'}
}
```

SPOT.TEST - Sends test contacts (spots) to the display.

```
data = {
        'type': 'SPOT.TEST',
        'params': {'AUTH': 'fixme'}
}
```

SPOT.SEND - Sends a contact (spot) to the display based on the furnished Zip Code.

```
data = {
        'type': 'SPOT.SEND',
        'params': {'AUTH': 'fixme',
        'SPOT': 'Location,PostalCode,97045,Oregon City,#ff2a00'}
}
```

SPOT.GRID - Sends a contact (spot) to the display based on the furnished Grid Square.

```
data = {
        'type': 'SPOT.GRID',
        'params': {'AUTH': 'fixme', 'CALL': 'T32ZA', 'GRID': 'BJ11LV', 'COLOR': '#ff2a00'}
}
```

CLOSE - Terminates GeoServer

```
data = {
        'type': 'CLOSE',
        'params': {'AUTH': 'fixme'}
}
```

Debugging

In the config file there are three parameters in the DEBUG section.

consolelevel is set to a value between 0 and 10, and is used to determine the verbosity of the messages written to the console window running the script.

logfilelevel is set to a value between 0 and 10, and is used to determine the verbosity of the messages written to the logfile.

logfile specifies the path and file name of the logfile where the debugging information is written.

The settings for the levels is described below.

0 - Logging is disabled

1 - LOG_SMDR, station message detail records. A short transaction summary designed for display in the console window, or consumption by your favorite log monitoring tool.

2 - LOG_SEVERE, severe errors that require script exit.

3 - LOG_ERROR, error messages that can't be recovered from but the script can continue to run.

4 - LOG_WARNING, recoverable problem that you should be notified about (e.g., invalid value in a configuration file, the program fell back to the default).

5 - LOG_INFO, informational messages.

6 - LOG_ENTRY, log entry and exit to all functions.

7 - LOG_PARM, log entry and exit to all functions with parameters passed and values returned (including global effects if any).

8 - LOG_DEBUG, general debugging messages, basically useful information that can be output on a single line.

9 - LOG_HIDEBUG, far more detailed debugging messages such as hex dumps of buffers.

10 - EXPERIMENTAL, used to implement experimental functions.

Each level also logs messages in 'lower' levels.

License

This script is licensed under the GNU GENERAL PUBLIC LICENSE, a copy of which is included in this distribution.

Change Log

Version 0.4 (November 2024)
- Added security authentication for commands.

Version 0.3
- Minor bug fixes and performance improvements.

Version 0.2
- Minor bug fixes and performance improvements.

Version 0.1 (March 2022)
- Original version.