

Introduction

The purpose of this document is to provide some details on how to create a logs on a map web page similar to the one shown below, a live version of which may be accessed from the Logs On A Map page of the CQ/X website (www.no5w.com). Reviewing that page before continuing with this document would be helpful and is recommended. The methodology uses AJAX, version 3 of the Google Maps API, CQ/X, and the GPS Log created by CQ/X.

The screenshot displays a web interface for logging QSOs on a map. On the left, there is a sidebar with a dropdown menu set to '2012 Kansas' and a list of 'Available Calls' including AA3OI, AA4GT, AA8IA, AB5ZA, AB5ZA/7, AB7RW, AC0BH, AC5O, AD0H, AD1B, AD1C, AE0G, AE7F, AI4SV, AI4VE, AK2U, DK2OY, DL3GA (highlighted), DL3IAC, DL3USA, DL6KVA, DL8USA, ES1TU, F5NKX, IN3NJB, JH8JYV, JO7WXN, K0CA, K0CCM, and K0DEQ. To the right of the sidebar is a list of QSOs for DL3GA, numbered 0 to 18, each with a call sign and '20M CW'. The main area is a Google Map showing a path of red and blue markers across Kansas and Oklahoma. The path starts in the north-central part of Kansas and winds south and east towards Oklahoma. A 'CQ/X de NO5W' label is visible in the top right corner of the map area. The map includes standard Google Maps controls like a compass, zoom in/out buttons, and a street view pegman. The bottom of the map shows 'Street View' and 'Map View' buttons.

In addition to a web server to host your logs on a map web page and the associated database, the following ingredients are required and will be described in detail in the following sections.

- A GPS log created by CQ/X
- A KML file created using CQ/X
- A GPS log modified to include a state-year code
- A MySQL database residing on your web server
- A PHP file installed on your web server and modified to access your database
- A HTML file installed on your web server and modified to include your Google Maps API key
- Four image files.

Step 1: Create a GPS Log

There is nothing for you to do in this first step since it is performed by CQ/X during your mobile operation provided that you are using a GPS and have it interfaced to CQ/X. The program will, by default, create a file containing the latitude and longitude of each of your QSOs along with other information. This file can be found in the following directory: C:\Program Files (x86)\CWSoft\CQxClient\Devices\GPS\GPS_Log\ and will have the following name: DefaultGPSLOG-YYYY-ContestName.txt where YYYY is the four-digit year and ContestName is the name of the QSO party. The file will be a comma separated value (CSV) file having the following format:

```
08/25/2012 14:50:02,From: N0R LAB (EM27fb),To: KG3W 14039KHz CW,37.0435,-95.5215
08/25/2012 14:50:59,From: N0R MGY (EM27fa),To: K8DSS 14039KHz CW,37.0416,-95.5336
```

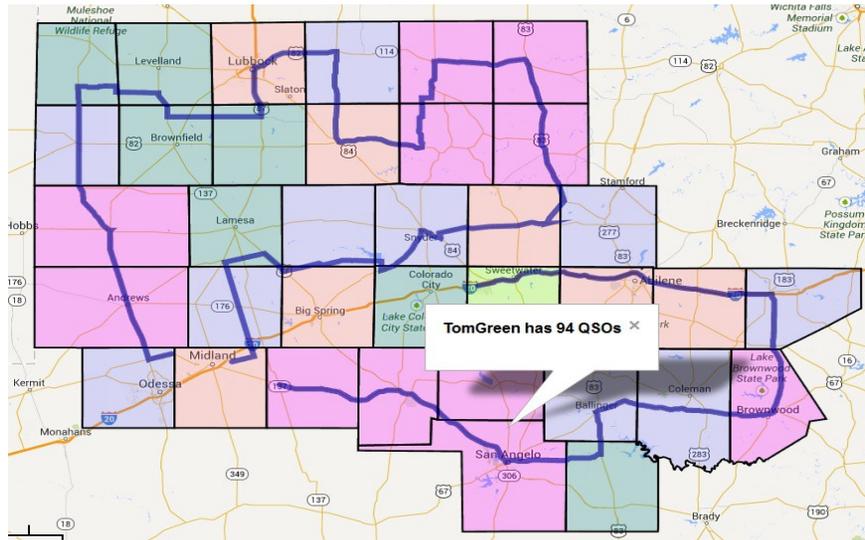
Note that the CSV fields are

- Date and Time in the format MM/DD/YYYY hh:mm:ss
- From field containing your call, county, and grid square
- To field containing the receiving stations call, frequency and mode
- Latitude field containing the latitude at which the QSO was started
- Longitude field containing the longitude at which the QSO started

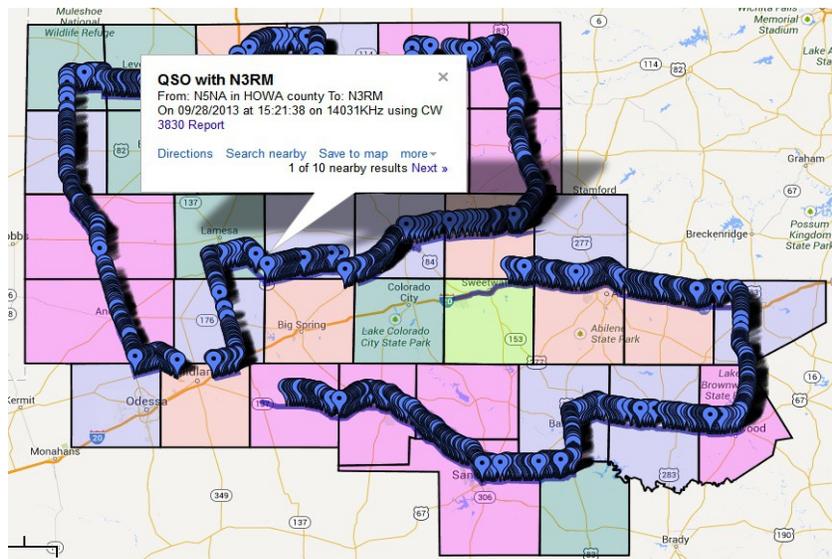
Step 2: Use CQ/X to Create a KML File for the Background Map

The KML file created in this step will serve as an underlying or background map on which the QSOs with each selected station will be highlighted. This KML file can be created using the menu item QP Tools | Prepare Log on a Map. Clicking on this menu item will bring up the dialog shown below. There are at least two options for creating this KML file.

Route Only Option: To implement this option uncheck the box labeled **Include All QSOs On Map** and check one or both of the boxes labeled **Include Route Trace Link** and then Browse to the file(s) that contain the network link(s) defining your route. This assumes that when you defined your route(s) using Google Maps you downloaded a small file that contains the network link. Typically you might have a route defined in Google Maps for Saturday and a separate one for Sunday or you might have a single route defined for the entire period of the QSO party. For this reason two network link files may need to be embedded in the KML. The background map produced using this option will have the following form showing only the counties, the route(s) taken, and the number of QSOs in each county – see figure on next page. No QSO placemarks will be present on this map. The QSO placemarks will be handled dynamically by the Google Maps API once the Log On A Map is published.



QSOs Option: To implement this option check the box labeled **Include All QSOs On Map**. You can also elect to include the Route Trace Links or leave them out by checking one or both of the Route Trace Link boxes but keep in mind that the QSO placemarks will essentially define your route on the resulting map so the Route Trace links may actually be redundant. If it is desired to include the Route Links follow the same process as in the previous option to specify the required files. The background map produced using this option will have the following form showing the counties, the location of each QSO, and the number of QSOs in each county – see figure on next page. The QSO placemarks shown in that figure will appear on the published Log On A Map and will be clickable. A separate set of placemarks showing all of the QSOs with a selected station will be dynamically presented by the Google Maps API when a given station is selected on the published map.



In both of the above options follow steps 1-10 described on the following dialog to create the KML file. It is recommended that you use a transparency/opaque value in the range 90-120 for best presentation of the county overlays. You will probably also want to simplify the name of the KML file after preparing it. If you find that your web server does not support a MySQL database and/or does not support PHP scripts you can, at this point, use the map produced under the QSOs Option as your log on a map by following the steps described in the document accessed using the Help | Posting Without Using GM API menu item. This will require you to create a web page for each contest. The resulting web page will not require the Google Maps API and therefore will not require the remaining steps described below but it will lack the interactivity that is possible using the API and each contest operation will need a separate web page.

Preparing KML for Log On A Map

To use this form to post your log on a Google Map you should use the Default GPS Log that was created for you automatically during the current contest. You can also browse to select a GPS log from this contest in a previous year. Perform the following steps.

1. Use the default or Browse to the GPS Log you plan to use.
2. Make sure that the GPS log is for the currently selected contest.
3. If not, change the contest and return to this dialog.
4. Specify what county overlays, if any, you want on the map.
5. If county overlays are being used specify the color scheme to use.
6. Select background and text colors for the QSO balloons.
7. Specify whether to include QSOs and/or Route Traces and the files for the Route Traces.
8. If desired, include a hyperlink in each balloon using the Find button to browse the internet to obtain the URL.
9. Click the prepare button to produce a KML file for posting on Google Maps.
10. If necessary use the How To button to access a document that explains how to post the map on the web.

GPS Log File:

There are 1441 QSOs in 36 counties. Max number of QSOs in a county is 70.

County Overlays

No county overlays
 Overlay counties in log
 Overlay all counties

Overlay Color Scheme

Counties with No QSOs	Counties with 1-10 QSOs
Counties with 11-20 QSOs	Counties with 21-30 QSOs
Counties with 31-40 QSOs	Counties with 41-50 QSOs
Counties with 51-60 QSOs	Counties with > 60 QSOs

County Boundary Color

Transparent Opaque

Folder Options

No folders
 County QSOs in Folders
 Station QSOs in Folders

QSO and Route Options

Include All QSOs On Map
 Include Route Trace Link
 Add'l Route Trace Link

QSO Balloon Options

Background Color Include Internet Link Label
Text Color

KML Data Will Be Written To

Directory: Show KML on Completion
As File:

Step 3: Modify the GPS Log to Include a State-Year Field

The database which will support the log on a map page will contain QSOs from all of the contest operations so it is necessary to be able to determine which operation each QSO belongs to. This is done

by adding a state-year field to the GPS log consisting of the two letter postal code of the state followed by the two-digit year of the operation. For example, when this is done, the two example lines given in step 1 become:

```
KS12,08/25/2012 14:50:02,From: N0R LAB (EM27fb),To: KG3W 14039KHz CW,37.0435,-95.5215  
KS12,08/25/2012 14:50:59,From: N0R MGY (EM27fa),To: K8DSS 14039KHz CW,37.0416,-95.5336
```

This step can be carried out using a spreadsheet. In a future CQ/X patch this state-year field will probably be added to the GPS log when it is created so before performing Step 3 please check to make sure that the state-year field has not already been added.

Step 4: Create and Populate a MySQL Database to Contain the GPS Logs

The MySQL database which will support the log on a map page should have the following two tables and their associated fields. Unless you plan to significantly modify the supplied PHP script be sure to use the exact names specified below for the tables and fields.

Maps table: This table should be named “maps” and should contain the following fields

- party – 4-character text field SSYY where SS is the state postal code and YY the two digit year
- mapfile – 255-character text field containing the full URL path to the KML file from step 2. So an example entry would be <http://www.xxxx.com/YourKMLFileName>

QSOs table: This table should be named “qsolog”. It contains the QSO information and should have the following fields.

- qsoparty – 4-character field SSYY where SS is the state postal code and YY the two digit year
- datetime – 20-character text field for holding date and time in the GPS log
- fromstr – 30-character text field for holding the “from” data in the GPS log
- tostr - 30-character text field for holding the “to” data in the GPS log
- latstr – 10-character text field for holding the latitude data in the GPS log
- lonstr - 10-character text field for holding the longitude data in the GPS log

Once you have created the database use whatever tools provided by your web host to populate the two tables. The maps table is typically quite small and can easily be manually populated but of course the qsolog table will be quite large and you will want to use some automated capability to populate it. I created a MySQL instance of the database on my local machine, populated it's qsolog table by importing a CSV version and then did a SQL export from that database and a SQL import on my hosted database since I could not get the CSV import to work on my hosted database.

Step 5: Modify the Provided PHP Script

The lines in the provided PHP script (GetLogOnMapData.PHP) that specify the database access parameters must be modified by inserting your hostname, username, password, and database name between the quotes in the lines at the top of the PHP file. These parameters are either supplied by your web host or defined by you when the database is created. Once the PHP file has been modified it should be copied to the home directory of your web site. The name of this file can be changed but if

you do so you will need to change the reference to the file in the provided HTML file.

Step 6: Modify the Provided HTML File To Include Your Google Maps API Key

To use the Google Maps API you will need to obtain a (free) key from Google and insert the key into the provided HTML file (DynamicLogOnAMap-Distribute.html) at the location indicated in the file.

```
<script type="text/javascript"  
  src="https://maps.googleapis.com/maps/api/js?key=InsertYourKeyHere&sensor=false">  
</script>
```

When obtaining the key be sure to obtain a key for version 3 of the API. Significant changes were made in version 3 of the API and the PHP and HTML depend on the functionality of that version. If you want to change the name of the PHP file you should modify the following line of the HTML file

```
var phpFileName = "GetLogOnMapData.PHP";
```

The name of the HTML file can be changed if desired. Once this file has been modified it should be copied to the home directory of your web site.

Step 7: Upload the Provided Image Files to Your Web Server

The following image files should be uploaded to the images directory of your web site.

- RightArrowWithCircle-red.png – image used to highlight the selected QSO
- RightArrowTransparent.png – transparent image used to un-highlight the non-selected QSOs
- Circle-Arrow-Red-1.png – marker image for showing QSOs with the selected station
- shadow-Circle-Arrow-Red-1.png – shadow of marker image used in showing QSOs

Adding a Log on a Map to existing Log on a Map Page – Once you have used the above process to create a Log on a Map web page it is easy to add logs on a map from additional operations and requires only a few of the steps from the above process as described below.

Step A1: Create a GPS Log – Of course this must be done at the time of the operations and as mentioned previously will be performed by CQ/X without requiring any action on the part of the user as long as a GPS is interfaced to CQ/X. So there is nothing to do here except to be aware of the location and name of the GPS Log.

Step A2: Use CQ/X to Create a KML File for the Background Map – Use the GPS Log and the CQ/X process described in Step 2 above to create the KML file. If desired, rename the KML file to a name consistent with the naming convention you are using in the “maps” table of the database and upload the KML file to your web server.

Step A3: Modify the GPS Log to Include a State-Year Field – Verify that the GPS log has a leading state-year field of the form described in Step 3 above. Use of a recent version of CQ/X will ensure this.

Step A4.1: Add a row to the “maps” table of the existing MySQL database

Step A4.2: Import the data from the GPS Log into the “qsolog” table of the existing MySQL database.

Steps 5-7 of the original process are not needed when adding a log on a map to an existing log on a map web page. Once the above steps are completed the new log on a map should show up on the page.