

GeoEvent Server / Waze / Cityworks Integration

The solution detailed below consists of three key components, the [Waze Connector](#), the [Event Volume Control Processor](#), and the Message Formatter Adapter. The first two items are add-on components available on the GeoEvent Gallery and GitHub respectively. The third component is included with GeoEvent Server but is not utilized by the default / preconfigured connectors. To use that adapter, you would need to create a new connector and attach it with the appropriate transport.

To simplify the deployment process, we have included the necessary compiled jar files for the add-on components, and provided a configuration file that has prebuilt the Message Formatter outputs. Unless already available in your system, please import the provided components and then load the configuration.

The screenshots below highlight the components of the service that you will need to modify:

Replace the Waze CCP Partner Name and Area of Interest with your approved values

The screenshot shows the ArcGIS GeoEvent Manager interface. The top navigation bar includes 'Monitor', 'Inputs', 'GeoEvent Services', and 'Outputs'. The 'Services' tab is selected. Below the navigation bar, the configuration for the 'waze-in-na (Receive Waze Connected Citizens Program Feed)' service is displayed. The configuration fields are as follows:

- Name*: waze-in-na
- URL: https://na-georss.waze.com/rtserver/web/TGeoRSS
- Waze CCP Token*: ccp_partner
- Waze CCP Partner Name*: REPLACE
- Data Types*: alerts
- Area of Interest: -118.451950589,33.8335267285;-118.059002277,33.8335267285;-118.05806
- Frequency (in seconds): 60

At the bottom, there is an 'Advanced' section that is currently collapsed.

Edit each of the provided outputs replacing the URL parameter with the services endpoint of your Cityworks server including a token in the following format:

https://cloud01.cityworks.com/MY_CITYWORKS_SERVER/services/AMS/ServiceRequest/create?token=e yJFbXBsb3llZVNpZCI6ODEwMywiSXNzdWVkbGltZSI6MTUwOTU0NjUwNzczNCwiTG9naW50YW1lIjoU1RSRUUVUyIsIlNpZ25hdHVyZSI6InhKekY4MnpWR2h4dnJobjd1MlVzcUJhbStiZ3dtQ1Fyd3EwOFVkdVnK2s9liwiVG9rZW4iOiJzaJqZTVTV1RBSVhEUpVZU0ycXo4L1VJb2wzUWdJVGNnbjNlIdm1YV2NFPSJ9

To generate that token, you will need to make an Authentication request from a browser directly to your Cityworks server's API. Here's an example of that request (taken from the Cityworks documentation):

```
http://localhost/Cityworks/Services/General/Authentication/Authenticate?data={
  "LoginName": "jdoe", "Password": "jdoe"
}
```

The screenshot shows the ArcGIS GeoEvent Manager interface. The top navigation bar includes 'Monitor', 'Inputs', 'GeoEvent Services', and 'Outputs'. The 'Outputs' tab is selected. The main configuration area is titled 'http-message-out-missing-signs-post (Push Message Formatter Output via POST)'. It contains several fields: 'Name' is 'http-message-out-missing-signs-post'; 'Message' is a JSON block with fields like 'ProblemSid', 'X', 'Y', 'CallerLastName', 'CallerComments', and 'Details'; 'URL' is 'https://cloud01.cityworks.com/_DEMO_KSMMS_15-2_LongBeach/services/AMS'; 'Use URL Proxy' is set to 'No'; 'Parameters' is empty; 'HTTP Method' is 'Post'; 'Header Parameter Name:Value List' is empty; 'Post/Put body MIME Type' is 'application/x-www-form-urlencoded'; 'Mode' is 'Client'; and 'HTTP Timeout (in seconds)' is '60'. 'Save' and 'Cancel' buttons are in the top right.

You can include additional service request types by creating copies of the outputs and modifying the ProblemSid parameter of the JSON Message block.

```
data= {
  "ProblemSid": "2641",
  "X": "${location.x}",
  "Y": "${location.y}",
  "CallerLastName": "Waze",
  "CallerComments": "Waze",
  "Details": "${subtype}"
}
```

By default, the service will accept any matching alert with a reliability of 2 or higher. If the volume of new reports is too high you can set the threshold higher (up to a maximum score of 10).



While not implemented in the provided solution, you could further reduce the number of provided service requests by connecting to the Cityworks Feature Service, create buffers around those features which can be written to a Stream Service, then used as Synchronized/Dynamic GeoFences to filter for new requests (e.g. not capturing service requests for potholes in the same area as open requests).