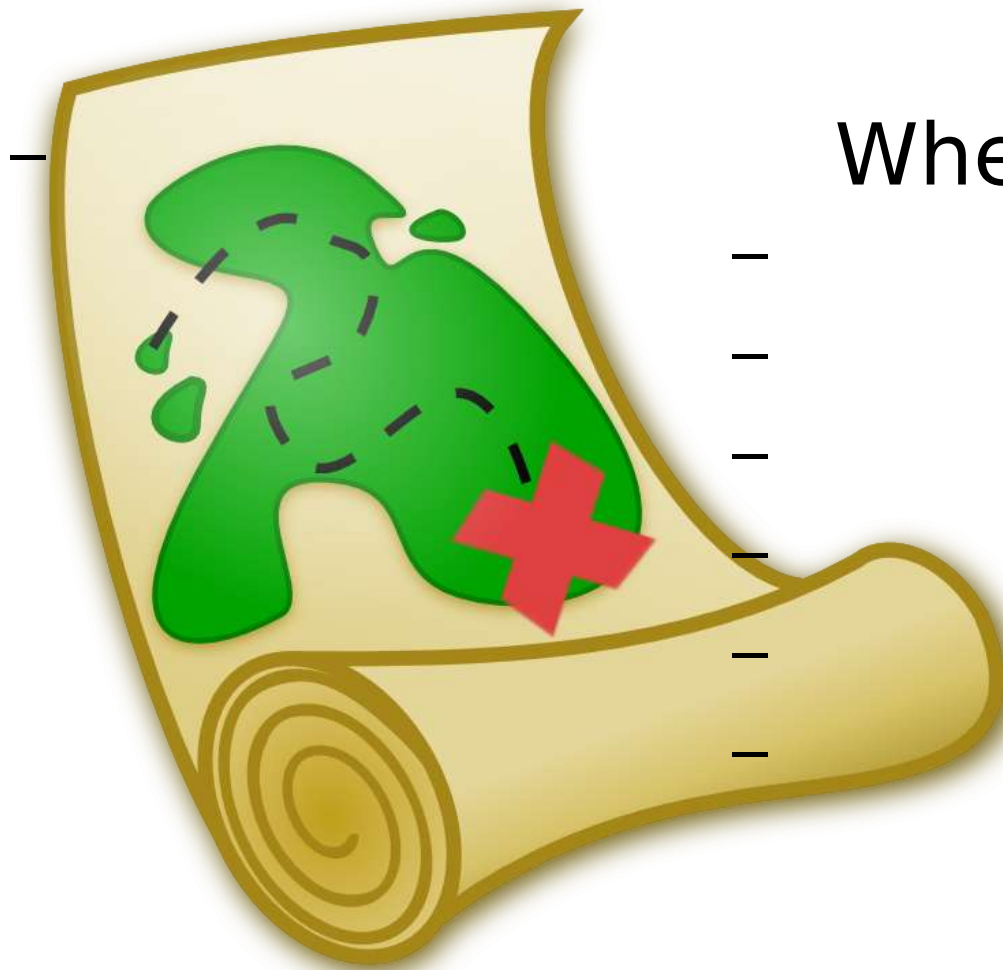


# GIS on Drupal in 2008



Where we ARRR

At the DrupalCon in Boston, we talked a lot about plans. This time, let's talk about what we can do *right now*.

We do mapping *right now* using the Location and GMap modules.

- Location is nearing a 3.0 release
- After years of purely dev releases, GMap 1.0 is in Beta

# The current iterations of both GMap and Location are for Drupal 5

- GMap for Drupal 6 is a functional 'dev'
- Location for Drupal 6 is... in progress

The Location module records,  
geocodes, and stores  
addresses.

The GMap module is an implementation of the Google Maps API for Drupal.

- Displays geospatial data on embedded Google Maps
- Provides controls for Google Maps attributes

# Things you can do with GMap and Location right out of the box

- Location version 5.x-3.x-dev
  - Has features that 5.x-3.0-test2 lacks
- GMap version 5.x-1.x-dev
  - No significant feature changes from 5.x-1.0-beta3

# User-submitted locations

- “where you live”
- “your favorite coffee shop”
- “drupal gatherings you've attended”



# User-submitted locations

- In this case, I allow anonymous users to create “user submitted place” nodes
- Anonymous users are also given the “submit latitude/longitude” permission so that they can use the “location picker” map widget provided by GMap.

# Display nodes on a map

- GMap provides a “Gmap View” view type, which we can use to build a map “view” of the user-submitted locations
- Different map markers are associated with taxonomy terms using GMap Taxonomy

# Filter a view by proximity

- A list of nodes with locations can be filtered by distance from a postal code

# Filter a view by proximity

- Imprecise, because the proximity calculations are based on geocoded postal codes that are loaded into the `{zipcode}` table
- Zip code data for eight countries is included with Location in the *databases* directory. You have to manually load it into your database to use the proximity search:

```
- mysql -u db_user -p db_name <  
  path/to/location/database/zipcodes.us.mysql
```

# Generate GeoRSS

- When any node with a location is included in an RSS feed, Location can provide its' data as GeoRSS.
- Choose the format on the node type config page—  
`admin/content/types/node-type`—from the *Locative information > RSS Settings* section.

# Create one-off maps

- Use GMap macros to display maps in content areas.
- Add the GMap Macro filter to one of your input filters (*after* any HTML filters)
- The GMap Macro Builder is a clicky interface for generating macro text; a full dictionary of available attributes is available at <http://mapedelic.org/documentation>

# But there is more to Location and GMap

- Both have APIs for developers

# GMap has the “GMap Array”

- Useful for generating maps programmatically
- Check out the GMAP-ARRAY-DICTIONARY.txt file in the GMap package for structure
- Render a GMap array to display with `theme( 'gmap' , $map_array )`



# A GMap Array

```
<?php
    $map = array(
        'id' => // id attribute for the map
        'width' => // map width in pixels or %
        'height' => // map height in pixels
        'latitude' => // map center latitude
        'longitude' => // map center longitude
        'zoom' => // zoom level
        'maptype' => // baselayer type
        'controltype' => // size of map controls
        'behavior' => array(), // various map behavior flags
        'markers' => array(), // array of points on the map
        'shapes' => array(), // array of shapes to overlay on
the map
    );
?>
```

# Filled in...

```
<?php
// a simple GMap array
$map_array1 = array(
    'id' => "my-map",           // id attribute for the map
    'width' => "100%",          // map width in pixels or %
    'height' => "400px",        // map height in pixels
    'latitude' => 41.9023,      // map center latitude
    'longitude' => -87.5391,    // map center longitude
    'zoom' => 7,                // zoom level
    'maptype' => "Map",         // baselayer type
    'controltype' => "Small"    // size of map controls
);

$output = theme('gmap', $map_array);
?>
```

# GMap 1.0 features

-“It's supported”

# Location 3.0 features

# location\_newapi()

- Yes, there is a new Location API; check for it with  
`function_exists('location_newapi')`
- Location 3.0 features, including the API, will be consistent between D5 and D6
- `hook_locationapi()`, `location_save()`,  
`location_load_location()`

# New architecture

- Location <-> Drupal object associations are stored separately from address and point data
- Location display is controlled by a template file
- Location fields are a custom Forms API element

# Locations and Drupal objects

- One table stores the address and the lat/lon
- Another stores the nid/vid, uid, or “genid”
- “genid” can be used by a module to associate locations with things that aren't nodes or users
  - Taxonomy terms
  - Private messages
  - Roles
  - ...

# Next Up

- Location is fine for points, but we want to play with points and lines, too
- Geographic calculations: typically “X within Y”, “Q within R km of Z”



# Geo.module

- Separating geographic storage from address storage
  - Who cares about addresses? Talking in State boundaries, GPS -tracked paths, etc.
- Allie Micka has been working on this

# Resources

- <http://mapedelic.org/blog>
- <http://drupal.org/project/location>
- <http://drupal.org/project/gmap>
- <http://drupal.org/project/geo>