

An example of a DL Maproom Widget with Google Maps UI

```

. . .
<form id="pageform" . . .>
  . . .
  <input class="dlimg . . ." name="bbox" type="hidden"/>
  <input class="dlimgts . . ." name="region" type="hidden"/>
  <input class="pickareal" name="resolution" type="hidden" data-default=".5"/>
  <input class="dlimg . . ." name="gmap" type="hidden"/>
  <input class="dlimgts . . ." name="gmapRect" type="hidden"/>
  . . .
</form>

. . .

<fieldset id="content" class="dlimage withMap">

  <link rel="iridl:hasFigure"
    href="/SOURCES/.UCSB/.CHIRPS/.v2p0/.monthly/.global/.precipitation/X/Y/fig+colors+-fig//T/last/plotvalue/">

  <div style="height:700px; width: 100%; margin:6px;">
    <div id="map1" class="dlimgGMap" style="height:80%; width: 98.5%;">
  </div>

  <script>

createGMapDLImage({

  id: 'map1',
  debug: false,
  mapOptions: { //https://developers.google.com/maps/documentation/javascript/reference#MapOptions
    center: {lat: 18.61168, lng: -71.373535},
    zoom: 3,
    fullscreenControl: true,
    styles: gmapStyles.iridl2,
    . . .
  },

  imageLayers: [
    { type: 'iridl',
      name: 'CHIRPS',
      opacity: 0.2,
      url: '/SOURCES/.UCSB/.CHIRPS/.v2p0/.monthly/.global/.precipitation/X/Y/fig+colors+-fig//T/last/plotvalue'
    },
    . . .
  ],

  mapClick: {
    type: 'marker', // none | click | marker | rectangle
    showFeature: true,
    moveOnClick: false,
    setBounds: true, // sets bbox and gmap when bounds change
    marker: { // google.maps.Marker options (without map)
      draggable: true,
      position: {lat: 18.0, lng: -71.0},
      . . .
    },
    rectangle: { // google.maps.Rectangle options
      bounds: { north: 20.0, south: 17.0, east: -69.0, west: -73.0 },
      editable: true,
      draggable: true,
    }
  },

  featureLayer: { //optional
    format: 'tsv', // tsv | geojson
    url: '/SOURCES/.NOAA/.NCDC/.GCPS/.MONTHLY/.STATION/IWMO/%282900100%29/%283909200%29/RANGE/table/:table/R.tsv',
    makeFeature: function (row) { // for tsv format
      var feature = defaultMakeFeature(row);
      feature.properties.name = row.Name;
      feature.properties.description =
        '<b>' + row.Name + '</b>,' + row.country_name + '<br>Elev: '+ row.elev + ' m';
      var elev = row.elev ? parseFloat(row.elev) : -99;
      feature.properties.elevation = elev;
      feature.properties.color = (elev == -99 ? "black" : (elev > 1000.0 ? "red" : "yellow"));
      return (elev == -99 ? [] : [feature]);
    }
  },
  featureClick: 'click', //none | click | infoWindow

```

```
makeMarkerOptions: function (feature){ // optional, returns marker options
  return {
    title: feature.getProperty('name'),
    icon: {
      path: google.maps.SymbolPath.CIRCLE,
      fillColor: feature.getProperty('color'),
      fillOpacity: .5,
      scale: 4,
      strokeColor: 'black',
      strokeWeight: 0.5
    },
  };
},
makeInfoWindowOptions: function (feature) { //function
  return {
    content: feature.getProperty('description'),
    pixelOffset: new google.maps.Size(0,0)
  };
},
markerCluster: { //optional
  gridSize: 50,
  maxZoom: 15,
  imagePath: 'https://developers.google.com/maps/documentation/javascript/examples/markerclusterer/m',
},
},
});

</script>


<br/><br/>


<br/>

<img class="dlauximg"
  Src="/SOURCES/.IRI/.MD/.IFRC/.CPC/.Merged_Analysis/.monthly/.v1201/.prcp_est/X/Y/fig-+colors+black+-fig//T/(Apr)/plotvalue+.auxfig/+.gif"/>

</fieldset>
```

DL Maproom Widget API

Option	Type	Default	Description
id	string	-	Unique id of Google Maps widget and corresponding GM div
debug	boolean	false	Debug mode turn on draws debug grid and debug output to the console
mapOptions	mapOptions	-	Map options https://developers.google.com/maps/documentation/javascript/reference#MapOptions
imageLayers	[imageLayerOptions]	-	Defines image layers, the last layer has the highest z-order
type	"iridl"	-	iridl' — only DL image are supported in this version
name	string	-	The name of the layer, this name is displayed on the map
opacity	number	-	Initial opacity of the layer
url	string	-	Ingrid url that will be appended with psdef and plotrange statements and "+.gif", X and Y grids are expected, make sure that this url doesn't result in animated gif
mapClick	mapClickOptions	-	Defines widget interactive features
type	"none" "click" "marker" "rectangle"	"none"	Type of user interaction: none — no additional interactive features will be created click — mouse click on the map propagates to the maproom via variable 'region' marker — a marker is placed on the map, when the user moves the marker, the position is propagated to the maproom via variable 'region' rectangle — a rectangle is placed on the map, when the user moves or resizes the rectangle, the bounding box of the rectangle is propagated to the maproom via variable 'gmapRect'
showFeature	boolean	false	Show the interactive marker or rectangle on start
moveOnClick	boolean	false	Move the interactive marker or rectangle to the mouse click point
setBounds	boolean	false	Set variable 'bbox' and gmap (and maybe 'region') when the bounds of the map change (map move and zoom)
marker	markerOptions	-	The interactive marker options
rectangle	rectangleOptions	-	The interactive rectangle options
featureLayer	featureLayerOptions	-	Defines the feature layer (for example, stations)
format	"tsv" "geojson"	-	The format of the input format
url	string	-	Features either in tab delimited format (tsv) or geojson

Option	Type	Default	Description
makeFeature	function (row) -> [feature]	-	When the input format is "tsv", this callback function will be called for each tsv row.. This function accepts an associative array (name: value) with one tsv row on input , and returns and array (possibly empty) of features
featureClick	"none" "click" "infoWindow"	none	Defines what happens when a user clicks on a feature: none — nothing happens click — mouse click on the feature propagates to the maproom via variable 'region' infoWindow — the infoWindow is opened
makeMarkerOptions	function (feature) -> markerOptions	-	This callback function will be called for each feature. Takes a feature on input, and has to return marker options. You may customize feature markers with this function, for example, color markers depending on mean precipitation
makeInfoWindowOptions	function (feature) -> infoWindowOptions	-	This callback will be called for a feature when infoWindow will need to be opened.
markerCluster	markerClusterOptions	-	Defines feature clustering. To switch off this feature, this element needs to be removed (commented out)