JSON data file specification for software package TRV

Tarmo Tanilsoo

Revision: 1

Date: 19.01.2016

Preface

Using the feature „Dynamic data points“(DDP) in the TRV software(beginning version 2016.1.11), it is possible to display custom data points within the radar plots. The initial rough implementation has since been replaced with the current version. These datafiles can be stored locally or on a web server for remote retrieval.

Metadata

A DDP JSON file contains value-key pairs, containing properties describing the file, as well as the data payload. The following is an example of a such file, indendation is provided for clarity purposes.

```
{
  "comment":"Any comment one may see fit to attach to a file. Not parsed by
software",
  "data":[{
          "color": "#ffffff",
          "icon":"home",
          "la":58.16,
          "label": true,
          "lo":26.45,
          "min":0,
          "size": 12,
          "txt":"Laguja",
          "type":0
          }],
  "name":"An example DDP data file",
  "interval":5
}
```

comment – string - Any comment that one may see fit to attach to a file. Not read by software at least at this point.
data – data payload. For more information, see below.
name – string. Name of a data file – shown also in the configuration window.
interval – minimal time interval in minutes between two successive downloads of the same file. The software would only reload the file if the minimum interval has passed since the last download. In the example, the file would not be downloaded again until five minutes have passed since the last download. This value is ignored on local files.

Data payload

The data payload consists of an array of objects, listed in the order of rendering. The exact content of these objects depends on the object type, described as value *type* in the object. The available object types are:

- 0 – Point
  - Object containing a marker, and optionally a text label.
- 1 – Line
  - Object containing a line with specified color and width.
- 2 – Label
  - A text label centered at a given coordinates.

Object type 0 – Point

icon – string - define the name of an icon to be displayed as a position marker. If set at *null*, standard position marker will be shown. The list of usable icons will be described at the end of the document.

la – float - latitude of the point in decimal degrees. Positive values are North. (58°N = 58.0)

label – boolean describing whether text should be displayed to the right of the position marker.

lo – float - longitude of the point in decimal degrees. Positive values are East. (26°E = 26.0)

min – float - minimum zoom level at which the object is displayed. To show the object at all zoom levels, set it at 0.

type – integer - type of object.

The properties above are mandatory. Following properties are required only if the property *label* has been set to *true*:

color – the colour of text associated with the object. Recommended format is #rrggbb where rr, gg, bb are R, G, B values in hexadecimal system. Also other formats valid in PIL/Pillow are valid as long as they are passed as strings. For example `"color":"rgb(255,255,255)"` would also be valid.

size – font size in pixels.

txt – text to be displayed.

Object type 1 – Line

conn – boolean – if true, the endpoints of the line will be connected.

color – string - the colour of line associated with the object. See also description for the property in object type 0.

la – array of latitudes. Latitudes are in the same format as in object type 0.

lo – array of longitudes. Longitudes are in the same format as in object type 0.

min – see object type 0.

type – integer - type of object.

width – integer - line width in pixels.

All properties are mandatory.

Please note – the plot uses azimuthal equidistant projection. As a result, please take related pitfalls into account.

Object type 2 – Label

color – see object type 0.

la – see object type 0.

lo – see object type 0.

min – see object type 0.

size – see object type 0.

txt – see object type 0.

type – integer - type of object.

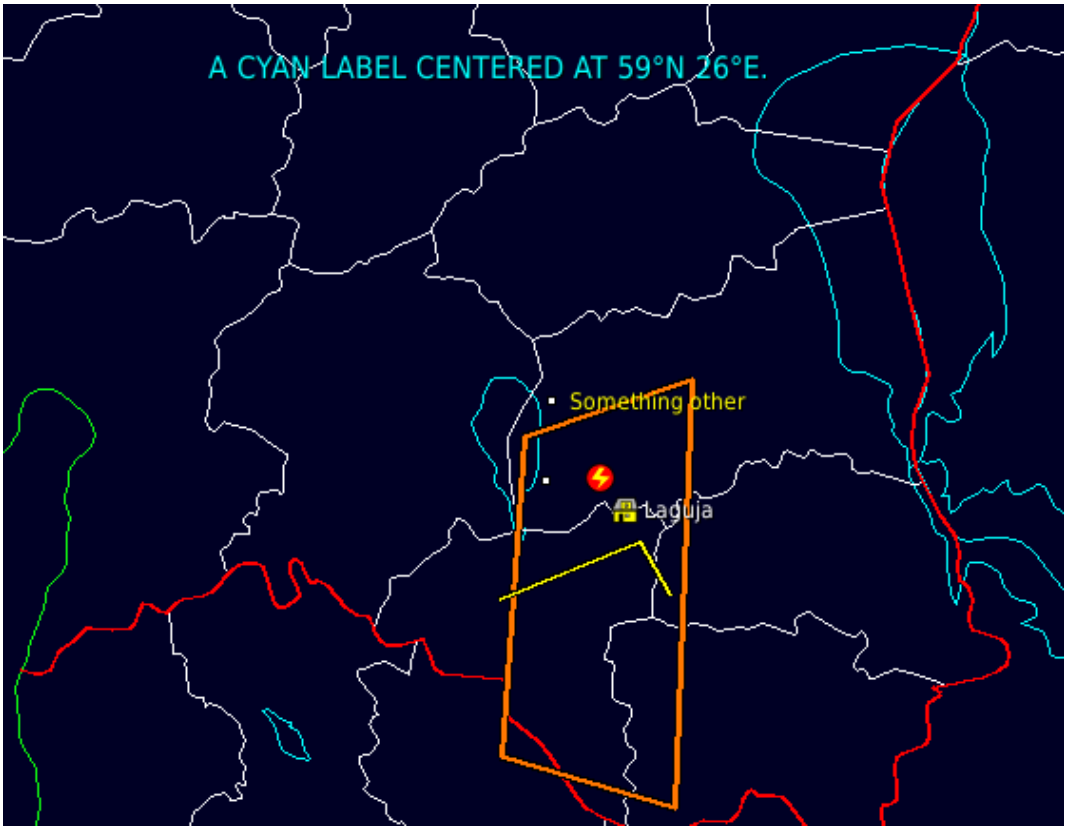
All properties are mandatory

Note: Unlike object type 0, displayed text will be centred at the given point.

Example JSON file and its output

```
{
  "comment":"Any comment one may see fit to attach to a file. Not parsed by software",
  "data":[{
          "conn":true,
          "color":"#ff7700",
          "la":[57.7,58.3,58.4,57.6],
          "lo":[26.0,26.1,26.7,26.6],
          "min":0,
          "type":1,
          "width":3
          },
          {
          "conn":false,
          "color":"#ffff00",
          "la":[58.0,58.1,58.0],
          "lo":[26.0,26.5,26.6],
          "min":0,
          "type":1,
          "width":2
          },
          {
          "color": "#ffffff",
          "icon":"home",
          "la":58.16,
          "label": true,
          "lo":26.45,
          "min":0,
          "size": 12,
```

```
          "txt":"Laguja",
          "type":0
          },
          {
          "color": "#ffff00",
          "icon":null,
          "la":58.37,
          "label": true,
          "lo":26.20,
          "min":0,
          "size": 12,
          "txt":"Something other",
          "type":0
          },
          {
          "icon":null,
          "la":58.22,
          "label": false,
          "lo":26.17,
          "min":0,
          "type":0
          },
          {
          "icon":"lightning",
          "la":58.22,
          "label": false,
          "lo":26.37,
          "min":0,
          "type":0
          },
         {
          "color": "#00ffff",
          "la":59,
          "lo":26,
          "min":0,
          "size": 16,
          "txt":"A CYAN LABEL CENTERED AT 59°N 26°E.",
          "type":2
          }],
  "name":"An example DDP data file",
  "interval":5
}
```

Built-in icons

- **backup**
- **car**
- **crosshairs**
- **cross**
- **dot (since version 2016.01.19)**
- **fog (since version 2016.01.19)**
- **hail**
- **home**
- **lightning**
- **radar**
- **rain (since version 2016.01.19)**
- **snow (since version 2016.01.19)**
- **sun (since version 2016.01.19)**
- **tank**
- **tornado**

  Icons are 16x16 px transparent PNG images that reside in /images/icons directory. The names of the image files are also used as icon names in json files. For example, "icon":"lightning" inside JSON refers to /images/icons/lightning.png . Beginning from version 2016.01.19 it is possible to also use larger/smaller images, but it is recommended to keep them at 16x16 px.