IdevioMaps

for QlikView

User Manual

Overview

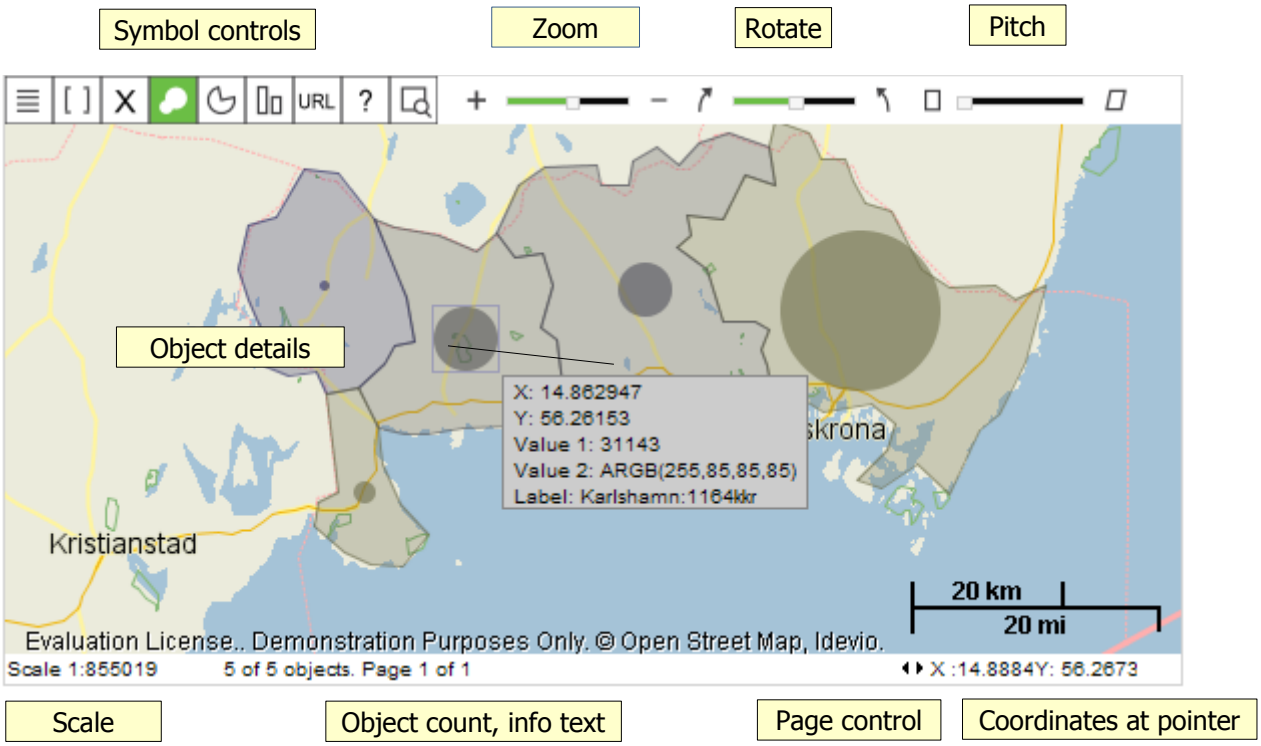
Idevio Maps for QlikView is an extension for QlikView version 10 and later. The extension enables advanced visualizations of business objects such as symbols, regions and tracks. This document describes how to use and configure the map extension. The main user interfaces are the map window and the map window properties.

Contents

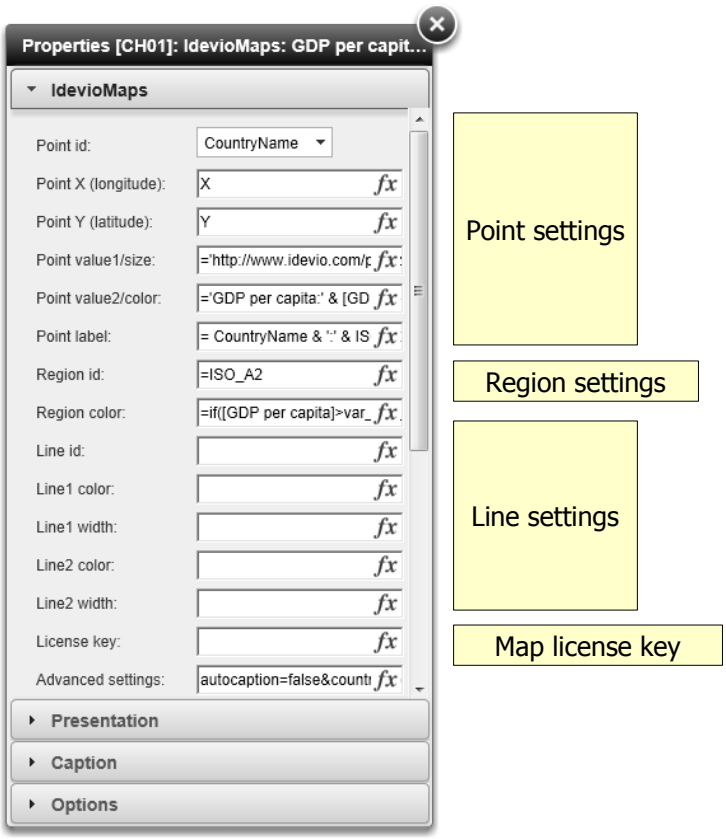
1 Overview

Idevio Maps for QlikView is an extension for QlikView version 10 and later. The extension enables advanced visualizations of business objects such as symbols, regions and tracks. This document describes how to use and configure the map extension. The main user interfaces are the map window and the map window properties.

1.1 Map window



1.2 Map Window Properties



2 Installation

The extension requires a valid key, please contact Idevio by web: *www.idevio.com* and use the web-based contact form or by phone *+46 31 779 09 60* to order.

Partners of Idevio may download a 30-day evaluation license from *www.idevio.com/qlikview*.

2.1 Quick start

This applies to QlikView desktop. See *section 2.2* on how to perform a server installation.

1 – Install the extension: double click on the qar files: IdevioMaps.qar and the IdevioAnimator.qar.

The qar file is a zip archive, QlikView will unpack and place the files in the extension folder:

Windows XP:
C:\Documents and Settings\user\Local settings\-Application Data\QlikTech\QlikView\Extensions\Objects\-Idevio\IdevioMaps
Windows 7:
C:\Users\ [UserName]\AppData\Local\QlikTech\QlikView\Extensions\Objects\Idevio\IdevioMaps

2 – Open the provided examples and verify that installation was successful.

3 – For each example insert the map license key. Right click on map window title bar and open properties, paste the key into the field "License key". Save the example and press F5 (refresh).

The map license key can be permanently installed in the extension, see section 2.4 Permanent install the map license key.

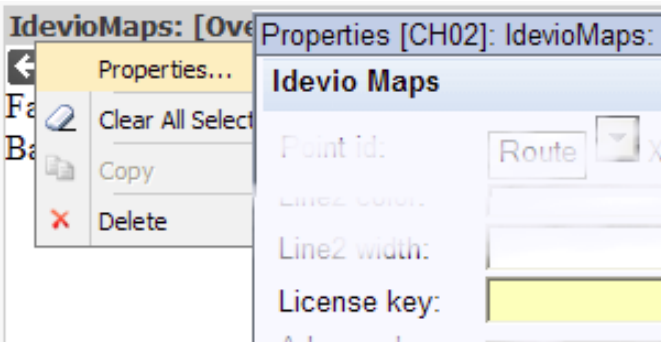


Illustration 1: How to enter the license key.

2.2 Quick start own data

1 – Read in data with a unique Id field and X, Y coordinates in WGS84.

2 – Turn on View → "Web View".

3 – Add map window to workspace: Right mouse click on background → New sheet object → Extension Objects → Drag the icon" IdevioMaps" to workspace background.

4 – Set Id, X and Y in map window properties: Right mouse click on map window title bar → Properties...

5 – Insert the provided map license key into the field "License key:". Save the application and press F5 (refresh).

2.2 Server installation

This applies to QlikView server. See *section 2.1* on how to perform a desktop installation.

Rename the IdevioMaps.qar to IdevioMaps.zip, unpack files and copy them manually to the Extension folder.

The folder varies per operating systems.

Ex: Windows XP, Server 2003:

C:\Documents and Settings\All Users\Application Data\QlikTech\QlikViewServer\Extensions\Objects\Idevio\IdevioMaps

Windows 7, Server 2008:

C:\ProgramData\QlikTech\QlikViewServer\Extensions\Objects\Idevio\IdevioMaps

Please note:

- The extension should reside on C: even if the Qlikview Server is installed on D:
- Make sure that the latest Java version is installed if documents are opened on the server.

2.3 Requirements

- QlikView v10 SR3 or later, v11 or later.
- Internet Explorer 8,9 when used together with QlikView Desktop. When the extension is deployed on the QlikViewServer the maps will work on most modern browsers including: Chrome, Firefox, Opera and Safari.
- Latest version of Java is recommended for best performance, with streaming maps. Java is free and available from www.java.com. Use IE 64-bit to download java if you are using Qlikview 64-bit.

2.4 Permanent install the map license key

The map license key is normally entered in the properties of each map window created. The license can be permanently installed by entering the license key in the Script.js file in the extension folder.

1 – Search the Script.js in the extension folder for: "Enter license key"

2 – Enter the provided license between the quotes.

Note 1: Be careful when entering the key, the file is compressed for speed and not easy to read.

Note 2: If the extension is updated the map license key must be entered again.

2.5 Locally installed map server

Normally the IdevioMaps for Qlikview runs towards the hosted map server at www.idevio.com. This can be changed for clients who run their own instance of Idevio RaveGeo Server WebMap. Switch map host by editing the Script.js file in the extension folder.

1 – Search the Script.js in the extension folder for: com.idevio.qlikview.ServerUrl="hosted.idevio.com"

2 – Enter the local hostname url (without protocol) between the quotes.

Note 1: Be careful when entering the url, the file is compressed for speed and not easy to read.

Note 2: If the extension is updated the url must be entered again.

2.6 Troubleshooting

Web view

QlikView needs to be set in “Web View”, extensions are only available in the QlikView Ajax client.

Administrator rights at install

Qlikview must have been installed with admin rights prior to extension installation. If not extensions might not be installed properly, it can help to copy the files manually. See section for server installation.

64-bit

If 64-bit versions of software is used, please make sure that all parts are using the same bit version: Qlikview, Internet Explorer and Java. The easiest way to get the 64-bit version of java is to use IE 64-bit and visit www.java.com, then the correct version will automatically be installed.

Qlikview "hangs"

If Qlikview should “hang” and show progress indicators in list boxes, it helps most of the time to refresh (F5). This can sometimes happen when changing the map properties.

Can't save workspace

If Qlikview should come in a state where the workspace cannot be saved, it helps most of time to reload the data. When an extension is removed and then added again this might occur.

Latest Java version

Unless otherwise stated we recommend the latest version of Java on client side for best performance and stability, visit www.java.com to obtain Java without cost.

Only 40 objects loaded

There is a bug in QlikView that limits the selection to 40 objects sometimes for extension at startup. Workaround: F5 refresh or update the selection. Or if the max objects is known, edit the Definition.xml in the extension folder, by adding a attribute to ExtensionObject tag, example:

<ExtensionObject PageHeight="3000">, however after that the number of objects loaded cannot be customized using the advanced settings.

3. Usage

3.1 Points

In order to visualize points the following are required:

- X,Y coordinates in WGS84 decimal degrees. Same as a normal GPS device delivers.
- Unique Id field (in order to make select from map to work)

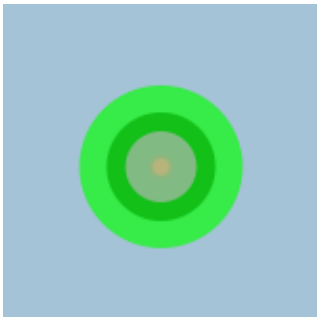
The dimensions are set in the map window properties.

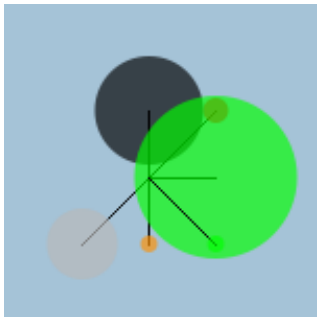
Symbols can be visualized with bubble, pie, bar charts or an arbitrary URL image. Switch between different presentations using the toolbar.

Point data can also generate a gradient heatmap, see separate section 5. Heatmaps.

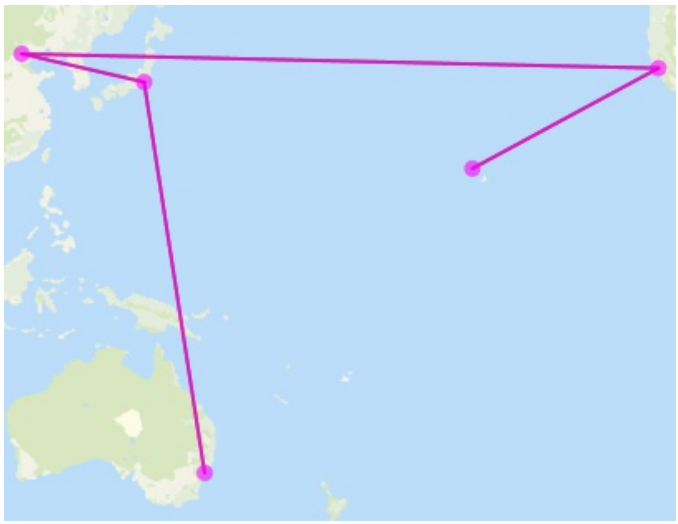
Numeric values are shown as a color ramp from blue to red. Text values are colored using a fix color table. Colors can also be defined using Qlikview function that returns RGB or ARGB functions. Non-transparent colors are treated as solid colors and given a default transparency.

If symbols are placed too close, the symbols can be separated. Lead lines will then indicate the original position when zooming in. Symbol separation is only available when Java is present on the device. Get Java free from www.java.com





Plotting beyond dateline and world wrapping. When in mapmode=html5 symbols and lines can be plotted beyond the -180 and 180 degrees in WGS84. The map will also wrap around when panned along the x axis.



To use the feature: to place symbols and render lines beyond 180 degrees, specify x as180+x+180. For instance Honolulu (-21.3, -157.816667) becomes (-21.3, 202.18333)

3.1.1 Labels

Labels on symbols are switched on and off using the toolbar. The label expression is set in the map window properties. If custom popups are used (see advanced settings) the first line is used as label.

3.1.2 Bubble chart

The default representation is a bubble icon where the first expression sets the size and the second sets the color. Shift to this symbol using the toolbar. Each symbol is resized to show the value relative to the max of the current selection. Pixel size is also possible and the bubble can also contain a label, see advanced settings.

3.1.3 Pie chart

Percentage values (0.0 to 1.0) can be shown as pie symbols, the first value expression sets the share, the second value the color. Shift to this symbol using the toolbar. The pie chart can show two values, using color as the second, see the advanced settings section. The pie chart starts at north and increase clockwise. The colors of the pie chart can be customized, see section about advanced settings.

3.1.4 Bar chart

Two numeric values can be shown as a bar chart. Shift to this symbol using the toolbar. The two values shares the same y axis, min and max are combined for the selection. The colors of the bar chart can be customized, see section about advanced settings.

3.1.5 Arbitrary image

Any GIF or PNG image can used as a symbol. Enter the URL to the image as the first value expression. Shift to this symbol using the toolbar. Using expression dynamic icons can be easily constructed, here is an example:

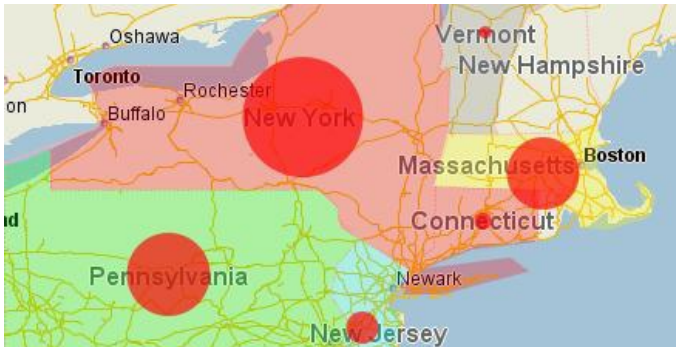
='https://chart.googleapis.com/chart?chst=d_map_spin&chld=0.35|0|fff|10|_|' & Position

3.2 Regions

Any region can be visualized in IdevioMaps, the regions are specified in a separate JavaScript file that is read in on demand. The color regions can be used for visualization values. Numeric values are shown as a color ramp from blue to red. Text values colored using a fixed color table. The region Id and the region color are set in the map window properties.

Regions and symbols can be combined and create a powerful presentation.

Here's a step-by-step guide, the JavaScript file is located in the same directory as Script.js.



1 – Get the geometry of the regions: id, polygons with x,y in WGS84. The id will become the key.

2 – Convert the regions to the following format, any custom regions can be used, syntax:

polys[i] = new ravegeo.Polygon('i', [list of x coordinates], [list of y coordinates], 'color in hex format use #00000000'); polys[i].regid = 'set to region id';

Example:

polys[0] = new ravegeo.Polygon('0', [-75.273744634000025, -75.273043610000002], [-14.04381441999999, -14.041436233000013], '#00000000'); polys[0].regid = 'A';

polys[1] = new ravegeo.Polygon('1', [-74.259776095000007, -74.234770982999976], [-14.441769765999993, -14.474387057], '#00000000'); polys[1].regid = 'B';

One region can consist of several polygons, it's important to specify the region id, otherwise the matching won't be correct.

3. Save the JavaScript file in the same directory as the extension and Script.js.

4. Start QlikView and add &regions_js=<file with polygons, ex: us_states.js> in the Advanced settings section of properties for map window.

See the example us_doctors.qvw, us_states.js and WorldRegionExample.qvw, world.js for full examples. Regions can be stored in hierarchy of js files and accessed from a region server, see next section.

Idevio provides a tool to convert from GML to region js file format: www.idevio.com/php/gml2webmap/

3.2.1 Drill down regions

Regions can be organized in a hierarchy to support drill-down. The js files containing the geometry for the regions can either be stored in local js files or served as a web service. By using index keys with '-' as the separator the correct region is read in. Idevio provides a web service with common regions hierarchies, set the advanced setting regions_js=http://geobroker-idevio.rhcloud.com/zipcodes.php? to enable it.

USA -> state -> county

Example of region key: USA, USA-CA, USA-CA-033. See example us_counties.qvw for complete list of codes for all states and counties.

Russia -> district - territory

Example of region key: RU, RU-Y, RU-Y-AD. See example us_ru_districts.qvw for complete list of codes for all districts and territories.

Germany -> 2 digit zip code -> 5 digit zip code area.

Example of region key: DE, DE-01, DE-01-01067. See example de_plz.qvw on how to use German PLZ codes.

Germany -> Bundesland -> Kreise.

Example of region key: de_bld_krs-15', de_bld_krs-15-

15086'. See example de_bld_krs.qvw on how to use German PLZ codes.

Great Britain -> Area -> District -> Sector postcode area

Example of region key: GB, GB-LL, GB-LL-LL99. See example gb_postcode_areas.qvw on how to use UK postcodes.

The Netherlands -> Provincie -> Gemeente -> Wijk -> Buurt.

Example of region key: NLD, NLD-DR, NLD-DR-M0106, NLD-DR-GM0106-WK010600. See example nl_admin_areas.qvw on how use Dutch administrative areas.

France -> Region -> Departement -> Arrondissement -> Canton -> Commune

Example of region key: FRA, FRA-93-04, FRA, FRA-93-04-041, FRA-93-04-041-0404. See example fr_admin_areas.qvw for complete list of codes for all states and counties.

Italy -> Regiona -> Provincia -> Comune

Example of region key: ITA, ITA-14, ITA-14-94. See example it_admin_areas.qvw for complete list codes of all regions.

Japan -> Prefecture -> Municipality

Example of region key: jp_pre_mun, jp_pre_mun-01, jp_pre_mun-01-01101. See example jp_admin_areas.qvw for complete list codes of all regions.

Local js file hierarchy

Regions can be stored in a hierarchy of js files, connected by the naming of the files. Heres the file name convention: <root name>-<key1>-<key2>.js. Here's an example:

world.js : all country geometries with country code as key.

world-US.js : all US states with state code as key

world-US-CA.js: all counties for California with county code as key

GeoJSON

In mapmode "html5" IdevioMaps can read regions defined with GeoJSON. The region geometries should be stored in WGS84, epsg code 4326. Each object in the region file should have a attribute "id" with value which serves as the index key. The region server can serve in GeoJSON format, use the following key: regions_js=http://geobroker-idevio.rhcloud.com/zipcodes.php?geojson

Data sources for drill-down regions

Natural Earth, US Census, PLZ - Seven, Geonames, Statistics Netherlands, European Environment Agency, IGN (Institut national de l'Information Géographique et Forestière), Istat (Italian National Institute of Statistics). Postal Boundaries © GeoLytix copyright and database right. Contains Ordnance Survey data © Crown copyright and database right. Contains Royal Mail data © Royal Mail copyright and database right. Contains National Statistics data © Geobasis-DE / BKG (Das Bundesamt für Kartographie und Geodäsie). Japan National Land numerical information, Ministry of Land, Infrastructure, Transport and Tourism.

3.3 Lines

Symbols can be connected by lines, and the lines are segmented by an id. Line id and line color are settings in the map properties dialogue. Lines can visualized in two modes: track lines and centric lines. Lines can be drawn as great circle arcs that follow the shape of the earth,

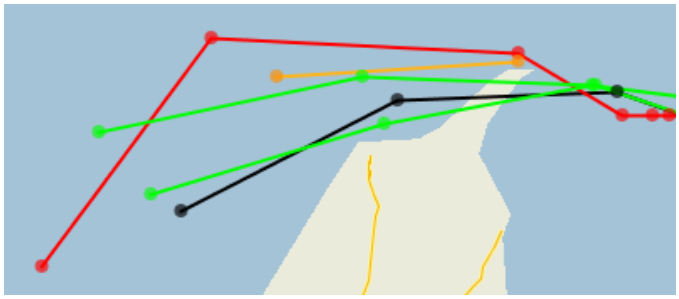
see advanced settings.

3.3.1 Line arrows

By default an arrow is placed on each line indicating the direction of the line. The arrow has the same color as line 1, with a darker tint. Arrows can be switched off, see the advanced settings section. If the mouse pointer is placed over an arrow, an pop up is shown with the line id and the object id.

3.3.2 Track lines

If the line color expression is a numeric value, the color is normalized based on max and min in the selection. If the value is a text value, the color is chosen from a color table. The color table can be customized in the Script.js. A workspace with Volvo Ocean Race is provided as an example.



3.4 Select

Select using the map can be done in several was ways:

- Select with mouse click on object
- Select visible objects with button in toolbar
- Select with the zoom rectangle tool.
- Select with the radius select tool.
- Select with the isoline select tool.
- Select with the polygon select tool.

3.4.1 Select with mouse click

Provided that a unique id has been chosen for points objects, a click on an object in the map will narrow the selection to that object.

3.4.2 Select visible objects

It is possible to narrow the selection using the map window by zooming in on a suitable area and objects and then press the “Select” button on the toolbar.

Zooming in and pressing “Select” again will narrow the selection further.

The “Clear” button in the toolbar will clear the current Id selection.

Note that the “Id” field in the map window properties must be set to a dimension with a unique field for the data set in order to make the selection through map window to work.

3.4.3 Select with zoom

IdevioMaps comes with a tool to make new selections by drawing a rectangle.

1. Click on zoom rectangle button in toolbar.

2. Draw rectangle around objects of interest.

3.4.4 Select with radius select

New selections can be done by drawing a circle.

1. Click on radius select button in toolbar.

2. Click to set the center of the circle.

The default radius of the circle is 6000 meters. This can be changed using the advanced setting “selectradius”.

3.4.5 Select with isoline select

IdevioMaps supports selections using an isoline. An isoline describing how far a car can travel in all directions given a maximum road distance or a driving time in minutes. Isoline requires a separate license, please contact Idevio to acquire one.

1. Click on isoline select button in toolbar.

2. Click to set the center of the isoline.

The default road length is 6000 meters. This can be changed using the advanced setting “selectradius”, driving time is set with the setting “isolinetime”.

3.4.6 Select with polygon select

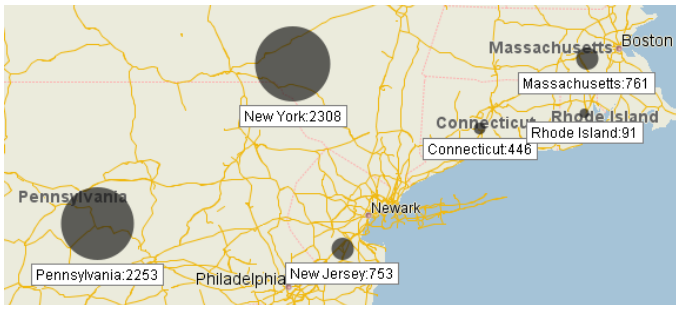
Another method for selection is using a polygon.

1. Click on polygon select button in toolbar.

2. Click to set points of the selection polygon. Place points so that the polygon contains the points of insterest.

3. Double-click to end the selection.

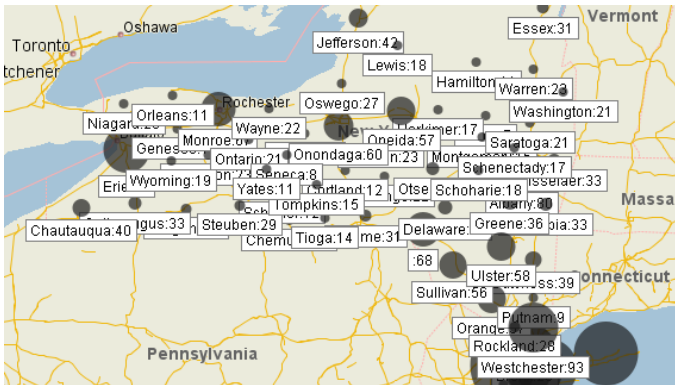
3.5 Drill-down



For large datasets that can be grouped, either hierarchic or cyclic, it's useful to use drill-down. The map supports drill down, use the group for “Point Id” and make expressions that calculate the average of the coordinates. How to:

1 – Create a drill-down group: Document properties → Groups → New.

2 – Set the newly created drill-down group as “Point Id” in map window properties



3 – Insert average of longitude and latitude, that will place the symbol in the right lace on aggregated levels. For example X: =avg(X) and Y: =avg(Y).

3.6 Advanced settings

The map state and advanced settings are kept in a special variable, visible in the properties dialog. Normally the map state is saved automatically and there is no need to change this variable manually. The advanced user can can edit and change settings that controls the map. The syntax is: *key=value&key2=value2*, same as arguments in a url, using '&' and '=' as separators. It's also possible to use '&&' and '==' as separators, if for instance an url with & and = including should be added as a value. The order of keys and values are not significant. The following keys and values are valid:

value=default | *options*

arrangescale=2000 *// Zoom level to start arranging overlapping symbols in a grid.*

autocaption=true|false| *// Controls if the title should be updated automatically or not*

barcol1=A2C180 *// Sets color of bar chart, value1, hex code: rrggbb.*

barcol2=3D7930 *// Sets color of bar chart, value2, hex code: rrggbb.*

***c_x**=11.7 // Default start center coordinate. The map will however zoom to the cover the area with the current selection.*

***c_y**=57.9 // Default start center coordinate. The map will however zoom to the cover the area with the current selection.*

charttype=0|1|2|3 *// Specifies symbol type, 0:Bubble, 1:Pie, 2:Bar and 3:URL image.*

citylabels=true | *false // Switch on and off city labels. Only available if java is present on the device.*

countrylabels=true | false *// Switch on and off country labels. Only available if java is present on the device. Get Java free from* www.java.com.

crs=4326|3006|3857 // Default coordinate system for background map is WGS84 EPSG code 4326., SWEREF99 (3006) is supported if custom map data in SWEREF99 is used. Web Mercator (3857) can be used in mapmode=html5 for external tile services.

custompopup=false | true // For customizing the popup window. Only the string from Point label is shown the pop up. Valid html tags: <b>

greatcirclearcs=false | *true // Whether centric lines should be straight lines or great circle arcs that follow the earth approximate curveness.*

gridheight=32 *// Specifies the grid spacing when arranging symbols.*

gridwidth=32 *// Specifies the grid spacing when arranging symbols.*

heatmap=false | *true // Whether symbols on the map should generate a multi color gradient heatmap.*

heatmapcolors=["#000000cc","#0000cc","#0000ff","#00ffff","#ffff00","#ff0000"] *// The default color ramp for heatmaps from low value to high value. Hex color notation #rrggbb or #aarrggbb. It's good to start with a transparent color for a nice transition.*

heatmapradius=25 *// The radius in pixels or meters of one point in the heatmap.*

heatmapopacity=0.8 *// 0-1, decides the transparency, 0 is completely transparent.*

heatmapunit=Meters | *Pixels // Whether the heatmaps should have geographic size in meters or in pixels.*

individuallinesegments=true | true *// Whether track line segments should be drawn with individual width and color or the same for the whole line. Lines with one color are faster to draw.*

***initscale**=3500000 // Default start scale. The map will however zoom to the cover the area with the current selection.*

***isolinetime=5** // Isoline select defined by driving time in minutes instead of meters.*

keepMapstate=false | true *// Controls if the map state should be read-only*

labelonsymbol= false | *true // Whether the bubble symbol should have the label printed inside the symbol or not.*

leadlinesvisible= true | *false // Whether arranged symbols should have a lead line or not.*

linedirectionarrow= true | false *// Whether lines should have and direction arrow or not.*

mapmode=auto|tiled|applet|html5 *// Force the map to be rendered with java or as tiled images or using html5.*

***maptheme=default**|tiledraster|empty // Options for html5 mode, tiledraster is suitable for slow computers, empty is no background map at all.*

maxrows=800 *// Max number of objects per page in*

selection.

minisymbolscale=30000000 *// Zoom level to switch on mini symbols.*

piecol1=ffffff *// Sets color of pie chart, value1, hex code: rrggbb.*

piecol2=A2C180 *// Sets color of pie chart, value1, hex code: rrggbb.*

piecolfg=3D7930 *// Sets foreground color of pie chart, hex code: rrggbb.*

piescale=false | *true // Scales the size of the pie chart symbol using Point val1.*

piescale=50 *// Size in pixels of pie chart if not scaled.*

pixelsize=false|*true // Size for bubble symbols is set in pixels.*

point_crs=4326|3006|3857 *// Default coordinate system for the data points.*

regionAsLineEnd=false | true *// If fields for regions id and color should be used as line end point x, y in centric line mode.*

regions_js=<js file with region polygons> *// Specifies which file to open and read any region polygons.*

roadlabels*=*true | *false // Switch on and off road labels. Only available if java is present on the device.*

selectonclick=true|false *// If select should be done object has been clicked.*

selectonzoomrect=true|false *// If select should be done after zoom rectangle operation.*

selectradius=6000 *// Sets the radius in meters of the select by radius and select by isoline tool.*

showlabels=false|true // If labels should be visible or not

statusbar=true | false *// Switch on/off the statusbar.*

symbolcontrols=true|false // If controls for symbol selection should be in toolbar or not.

symboldefault=10 // Default bubble symbol size in pixels

symbolmax=80 // Maximum bubble symbol size in pixels

symbolmin=5 // Minimum bubble symbol size in pixels

tabletmode=false | true *// Gui optimized for ipad and tablets with bigger buttons etc.*

toolbar=true | false *// Switch on/off the toolbar.*

tmsurl="" *// Available in mapmode html5, url to external tile services. Example: http://otile1.mqcdn.com/tiles/1.0.0/sat/{z}/{x}/{y}.jpg*

tmsurl2="" *// Available in mapmode html5, an optional 2nd tilelayer, see tmsurl.*

tmstilesided="152.874053955078" // The width of the smallest tile. Only in mapmode html5.

tmstop="-20037508.3428" // Top position of the smallest tile in tile ref system. Default for a global web mercator tile set. Only in mapmode html5.

tmsleft="20037508.3428" // Left position of the smallest tile in tile ref system. Default for a global web mercator tile set. Only in mapmode html5.

tspmax=12 *// Sets the maximum number of points that can be used for TSP shortest path calculations.*

twopievalues=false | true *// If a second value should be used instead of color for pie chart image*

zoomrect=true | false *// If zoom by rectangle shuld be enabled.*

zoominscale=1000 *// Zoom level to zoom to when clicking on an object.*

wmsurl="" *// URL to WMS (Web Map Service) image that will be loaded as a map backdrop when interaction is ended.*

3.7 Map control

3.7.1 Zoom

Use the “+” and “-” buttons or the slider in the toolbar. Zooming can also be performed by moving the mouse up and down while press and hold right mouse button. The mouse wheel can be used to zoom as well. Double click on the map (not on a object) will also make the map to zoom in.

Zooming can also be using the Zoom rectangle tool, point click and drag a rectangle over the area of interest.

3.7.2 Pan

Pan by moving the mouse while pressing the left mouse button on the map. Arrow buttons in the toolbar can also pan the map

3.7.3 Rotate

The map can be rotated using the rotate slider in the toolbar. Convenient to make the selection fit better the map.

Rotation is only available if java is present on the device. Get Java free from www.java.com.

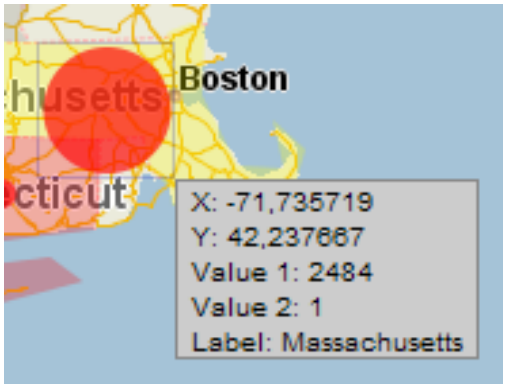
3.7.4 Pitch

The view of the map can be shifted from straight above to a perspective view using the slider in the toolbar.

Pitch is only available if java is present on the device.

3.7.6 Mouse over details

Moving the mouse pointer over an object will make a popup window with more details appear. The window contains values of the dimensions and the expressions



used in the map, or only the label string, see advanced setting *custompopup*. The dimensions and the expressions are defined in the map window properties.

3.7.7 Scale/Map window width

The status bar contains, to the left, an indication of the current scale. Clicking on the scale will show the current width of the map window.

3.7.8 Page control

The status bar contains, to the right, a page control which makes it possible to scroll very large selections. Use “<” and “>” buttons in the status bar to browse the selection.

3.7.9 Background map

The default background map is based on OpenStreetMap data, © OpenStreetMap contributors, CC-BY-SA.

Other map data is available on request. Idevio can supply maps from Tom Tom, Here, Lantmäteriet and many others sources.

3.7.10 WMS image as map background

IdevioMaps can show images from WMS (OGC Web Map Service) as an alternative map background. The WMS will be loaded as an image when the zooming and panning of the map has ended. The URL to the WMS source is specified in the advanced setting "**wmsurl"**. The URL should be complete but without the BBOX, WIDTH and HEIGHT argument. Example:

&&wmsurl==http://test.idevio.com/ravegeo/map2/WMS?LAYERS=_All_&FORMAT=image/png&NAME=WorldExample&REQUEST=GetMap&VERSION=1.1.0&BGCOLOR=0xFFFFFF&TRANSPARENT=T&SERVICE=WMS&STYLES=&SRS=EPSG:3395

Recommended (supported) SRS's are EPSG:3395, EPSG:3857, EPSG:900913. EPSG:4326 works in larger scales but will distort when zooming out.

Note 1, WMS as a map background is only available in the non-java version of IdevioMaps, mapmode=tiled.

Note 2, the WMS URLs often contains '&' and '=', please use the alternative separators in advanced settings: '&&' and '=='.

Note 3, WMS cannot be combined with gradient heatmaps.

3.7.11 External tile sets

External tile set providers can be used in IdevioMaps when in mapmode html5. Use the advanced setting tmsurl, tmsurl2 to specify source.

Example of providers:

MapQuest aerial tiles: http://otile1.mqcdn.com/tiles/1.0.0/sat/{z}/{x}/{y}.jpg

MapQuest map tiles: http://otile1.mqcdn.com/tiles/1.0.0/map/{z}/{x}/{y}.jpg

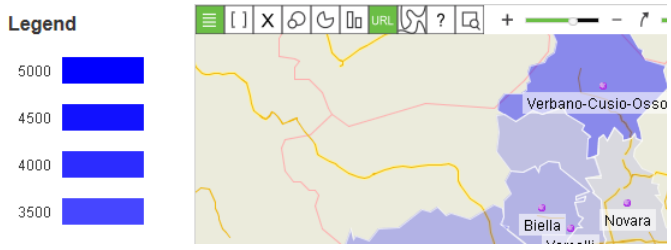
Stamen watercolor: http://tile.stamen.com/watercolor/{z}/{x}/{y}.jpg

The external source must be defined in Web Mercator, EPSG code 3857 and follow the TMS url convention with zoom level z, x, y in url. See example IdevioMaps_different_map_backgrounds.qvw for usage.

WMTS sources can also be specified, currently EPSG code 3010, 3011 and 3006 is supported. Use advanced setting tmstilesized, tmstop and tmsleft to specify tile size and origin.

3.7.12 Legends

There is no internal legend object map, however it's easy to create a chart in QlikView that can serve as a legend, see for example see it_admin_areas.qvw. Choosing consistent colors, a.k.a “Silent legend” in the QlikView document is an excellent best practice.



4. Scatter chart support

For clients that are restricted to older versions (v8, v9) of Qlikview we provide a way to embed a map as background to scatter charts. The interaction in a scatter charts is more limited compared to the extension, as to zooming, panning and symbol options. This map service

does not rely on the Qlikview extension API and can be used for OEM solutions and for the IE plugin.

Usage:

Variables for the axis*: var_xscale_min, var_xscale_max, var_yscale_min, var_yscale_max*

Variables for the backgrund: *var_mapurl*

1 – Acquire a license and example code from Idevio, use web form at www.idevio.com.

2 – Add scatterplot.

3 – Copy from example workspace load script, section "IdevioMaps" start to end.

4 – In load script:

– Replace x,y with your longitude and latitude.

- **Set scatterplot window width and height.**

5 – In scatterplot:

– Set background to =var_mapurl.

– x,y axis max and min to var_xscale_min, var_xscale_max, var_yscale_min, var_yscale_max.

– Unselect 'force 0'.

- **Set expressions for x,y and z.**

6 – Done!

5. Heatmaps

IdevioMaps offers an optional visualization for point datasets. Based on the value of Point value1/size a multi color linear gradient layer can be displayed.

1 – Add point data set, set Point Id, X and Y .

2 – Set Point Size to a numeric value. Higher values will have stronger impact on the heatmap.

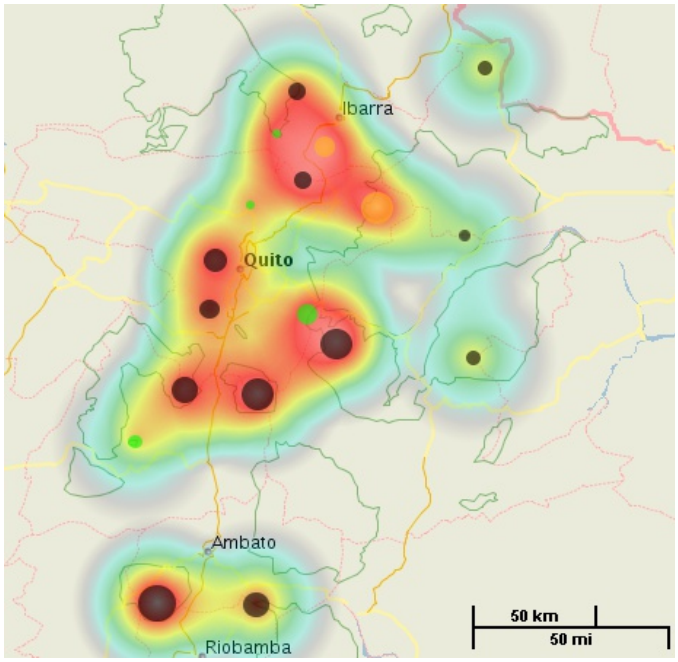
3 – Press the heatmap button in the toolbar.

The heatmap is normalized based on max and min in the selection and of the points visible in the map window.

The point radius, heatmap opacity and the heatmap units can be customized, see advanced settings.

Please also see the example Volcanoes_heatmap_example.qvw for demo of the functionality.

Heatmaps are only available in mapmode auto, tiled and java, ie not in html5 yet.

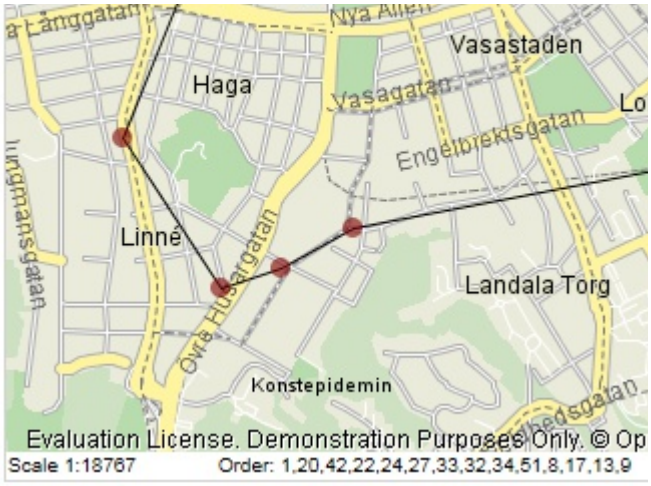


6. Route optimization

IdevioMaps can calculate a shortest path between points. The distance between points can either be bird distance or the actual road distance.

The default max number of points is 12 for shortest path calculations, if a high end client computer is used this can be increased using the advanced setting “tspmax”.

6.1 Shortest path, bird distance



Usage:

1 – Select the points of interest to find the shortest path in between, limit this to 12 points.

2 – Press the toolbar button “TSP” for a path based on bird distances.

3 – See the suggested path on the map, black line between points. The status bar contains a list of the order of points.

6.1 Shortest path, road distance

Actual road distances requires a special license from Idevio, please contact sales@idevio.com to purchase such a license.

1 – Select the points of interest to find the shortest path in between, limit this to 12 points.

2 – Press the toolbar button “TSP r” for a path based on road distances.

3 – See the suggested path on the map, black line between points. The status bar contains a list of the order of points.

4 – Click on statusbar to open a separate window with directions in text.

Hovering over the line and the breakpoints of the line will show popup messages with driving directions.

7. Animation

Sometimes it can be useful to animate data in QlikView. For instance show how a model varies over time, the hours of the day, the months of the year or just stepping through data types.

Idevio provides a way to do that through a separate extension the 'IdevioAnimator'. The IdevioAnimator is not linked explicitly to the map extension, it operates independently and steps through a selection.

Usage:

1 – Install the extension by double-clicking on the IdevioAnimator.qar file located in the base directory.

2 – Turn on View → "Web View".

3 – Add IdevioAnimator to workspace: Right mouse click on background → New sheet object → Extension Objects → Drag "IdevioAnimator" to workspace background.

4 – Set dimension to iterate over in extension object properties: Right mouse click on extension object title bar → Properties...

5 – Optional, set "Period" the time in ms to pause, set "Steps" number of steps to animate.

Please see examples for usage of IdevioAnimator: VolvoOceanRace2005.qvw, se_drilldown_houseprices_scb.qvw and IdevioMaps_CrimeStats_Chicago.qvw.

8. Map modes

IdevioMaps is capable of showing maps using different technologies. Map mode is controlled using the advanced setting 'mapmode'. The current map modes are: auto, tiled, java,html5, default mode is auto.

auto, tiled, java

IdevioMaps can render maps on client side, this gives streaming map with smooth zoom and pan. Another option is to render maps as tiles of images, the tiled mode.

With auto mode IdevioMaps detects if Java is installed and then starts in java (streaming maps) mode and in tiled mode if not. Tiled and java mode forces the map to start with images and with the streaming maps.

html5

This mode renders streaming maps using html5. A modern browser is required, ie the latest version that Qlik supports of Chrome, Firefox and Internet Explorer. The benefits of using this mode is faster startup time, printing is better, same appearance on phones and tablets.

8. Examples

Some of the examples provided with the extension are also available as web demos at www.idevio.com/demo.

Eu.qvw, City statistics from Eurostat

This demo visualizes urban living in a number of European cities. The statistical data from Eurostat 2007-2010. The example demonstrates how point data can be visualized using bubble, pie, bar and arbitrary images.

Us_doctors.qvw, US state statistics

The example shows region coloring with data from the National Center for Educational Statistics 2011. The regions are defined as US state polygons, with the state abbreviation as the key.

VolvoOceanRace2005.qvw, Tracks from Volvo Ocean Race 2005

This example shows line coloring using data and tracks from Volvo Ocean Race 2005, leg 8 and leg 9. The lines are visualized as tracks, connecting observations of boats to lines to colored.

FlightStats.qvw, Ryanair flight from UK statistics during 2010

The example shows the second line mode, centric lines, with lines for each object is drawn from a center point. The data is sourced from www.flightontime.info and shows delay statistics for RyanAir flights from UK during 2010. Each link is drawn with two lines, width depends on expressions. Note: start airport must be in selection to show lines.

Earthquakes.qvw, scatter chart maps example.

This demo shows IdevioMaps as a background for scatter charts. Suitable for very large datasets with little need for interaction and for solutions based on Qlikview 8 and 9. Data from USGS Centennial Earthquake Catalog.

WorldPostcodes.qvw, drill-down example.

This demo shows how to use IdevioMaps for drill-down datasets. Suitable for very large datasets that can be grouped, either hiercal or cyclic. The data, approx 900 000 zip codes, comes from geonames.org.

More examples are provided in the evaluation kit.

9. Idevio Maps extension details

9.1 Architecture

The IdevioMaps extension is built as a connection between the web based map engine from Idevio: RaveGeo WebMap and QlikView. The map data is streamed from a remote server and rendered by the client.

QlikView calls the extension if the selection has changed. The call contains an object that holds all objects in the current selection. At the first call the map window, the toolbar and the status bar is instantiated.

Objects can be shown as bubbles, pies or bar diagrams or an arbitrary image. Regions can be visualized if geometries and id keys is provided. Lines can be displayed between symbols if a line id is provided. The size and the color of symbols, regions and lines are normalized based on max and min in the selection. If the values are not numeric but text values, the color values are picked from a color table. The Qlikview color functions can also be used to set the color.

The map window will automatically zoom out so that the whole selection is visible in the map window.

The map component has a JavaScript interface with dual implementation. If Java is present on the client, maps are rendered on the client, otherwise the maps are shown as server side rendered images.

9.2 File structure

The extension consist of the following files:
Definition.xml *// Defines the number of dimensions and expressions used by the extension.*
Properties.qvpp *// Defines the look of the property dialogue.*
Script.js *// Main file, defines the behavior of the extension.*
jquery-ui-1.8.9.custom.min.js *// jQuery UI*
jquery-ui-1.8.9.custom.css *// jQuery UI, toolbar and statusbar*
jquery-1.4.4.min.js *// jQuery, toolbar and statusbar*
us_states.js *// Example file for region coloring*
Icon.png *// Icon used by New sheet object... dialogue*

The javascript files are compressed for higher

performance.

10. Hints on extension development

- QlikView returns first all dimensions then all expressions in this.Data.
- F5 in QlikView Desktop will reload the extension, handy when editing Script.js
- If the Definition.xml has changed, the map window must be removed and added again. Qlikview can't handle extensions in multiple versions.
- Select on several dimensions simultaneously don't work.
- Any resources (CSS, JavaScript, images etc) in the extension folder Idevio/IdevioMaps must be in root, QlikView web server can't address sub-folders.
- The map window properties dialog in version v10 has a fixed size and cannot be resized.
- Qlikview v11 has new set of gui components and needs custom made qvpp file and uses jQuery.

11. Links to further documentation

11.1 Idevio RaveGeo WebMap

The extension is based on Idevio's web based map engine: RaveGeo WebMap2 and WebMap5: http://developer.idevio.com.

11.2 jQuery

jQuery is used for customize the map extension with a toolbar and a status bar.

www.jquery.com

www.jqueryui.com

11.3 QlikView JS API

The documentation for QlikView extension JavaScript API is available from the start menu when the Qlikview server is installed.

Stephen Redmond has written a nice intro to QlikView extensions:

http://qliktips.blogspot.com/2011/01/beginners-guide-to-qlikview-extension.html

http://qliktips.blogspot.com/2011/01/beginners-guide-to-qlikview-extension_09.html

http://qliktips.blogspot.com/2011/01/beginners-guide-to-qlikview-extension_4612.html

http://qliktips.blogspot.se/2011/04/beginners-guide-to-qlikview-extension.html

12. Change log

Ver 4.1.24797: Option to hide/show toolbar and statusbar.

Ver 4.1.24663: WMTS support through TMS, with tile side, top left. World wrapping of symbols and lines, plotting beyond dateline.

Ver 4.1.24344: First version with a new map mode html5. Heatmap custom color ramp. Default settings for center and scale. Swedish postal code areas and Japanese administrative boundaries included.

Ver 3.1.23778: Route optimization: shortest path between points, bird and road distance. New selection tools with radius, isoline, polygon. Max rows bug fixed.

Ver 2.3.23342: Animation control of dimension, label inside bubble symbols, scalable pie chart symbols.

Ver 2.3.23235: Gradient heatmaps now in streaming version. Fixed qv bug at exit/switching tabs. Sweref99 support.

Ver 2.3.22968: Multicolor linear gradient heatmaps and WMS backdrops. Alternative separators in advanced settings. New examples.

Ver 2.3.22851: New drill down region examples, Italian and French municpalities. Bug fixes and improved error handling for regions.

Ver 2.3.22806: Popup windows can be customized with html tags. First line as label.

Ver 2.3.22778: City and road labels can be switched off. Leadlines in symbol arranging can be switched off. Grid size in symbol arranging can be customized. Bubble symbols can be specified in pixels. Transparency can be set for regions, lines and symbols. Edges on regions are now white. Bug fix for resize problem in Internet Explorer. No rezoom after symbol change. No rezoom after label on/off

Ver 2.3.22617: Hierarchical region files, local or from web service.
Ver 2.3: Demo design updated**.**

Ver 2.22518: New design of IdevioMaps, toolbar, demos and manual.

Ver 2.22471: Qlikview v11 specific prop page. Select on click and zoom rect. Bar and pie chart custom colors. Bugfix for IE9.

Ver 2.22193: Bugfix for IE7, toolbar not visible, sliders now v11 look.

Ver 2.22066: New example, countries of the world as regions. Semitransparent labels. Country labels on/off. Popup dialogs on click.

Ver 2.21920: License key now in properties dialog

Ver 2.21830: Verified on Qlikview v11, regions specified in advanced settings.

Ver 2.21773: Fix for invalid characters in region id.

Ver 2.21749: Great circle arcs and width and color on line segments.

Ver: 2.21744: Upgraded to jQuery 1.7, jQuery UI 1.8.9, Fixed IE7 bug, Removed pitch and skew slider in tablet mode, Nicer error messages.

Ver: 2.21622: Enabled drill-down, replaced dimensions with expressions.

Ver: 1.21173: Added support for Qlikview ARGB functions.

Ver: 1.20928: Added arrows on lines.

Ver: 1.20832: Improved region handling

Ver: 1.20819: Added error handling on dirty x,y. Select on zoom.

Ver: 1.20574: Zoom rectangle tool, more settings, default rows and scales.

Ver: 1.20519: Added centric line mode with end point in column.

Ver: 1.20353: mapstate: saves map state in qv doc, keepMapstate: makes map state readonly, autocaption: custom title on/off, tabletmode: bigger buttons etc for ipads, linemode star: lines originating from center points. RGB function enabled in expressions. Version number on load and about. Same scale for bar and line widths.

Ver: 1.19731: Expressions for region color, local js files for regions.

Ver: 1.19662: Line support.

Ver: 1.19563: Multiregion coloring, opt map license key in properties.

Ver: 1.19520: Area coloring.

Ver: 1.19504: Expressions for value and label.

Ver: 1.19349: Added page control

Ver: 1.19083: Statusbar added.

Ver: 1.18392: Added jquery, jquery ui