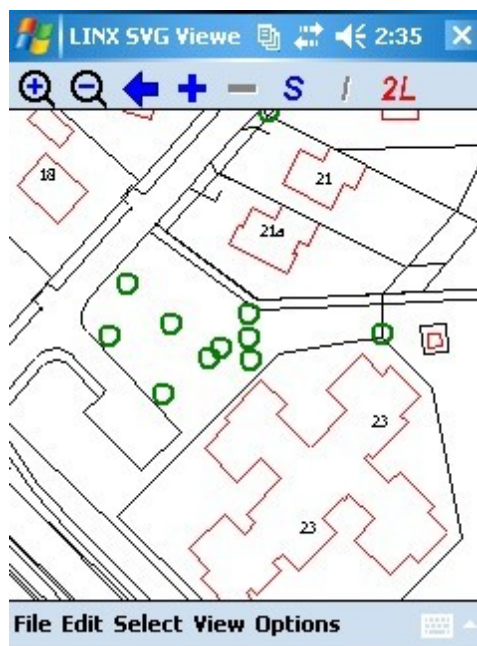


Mapping

LINX SVG Viewer uses the tiny SVG (Scalable Vector Graphics) W3C standard as basis for its SVG maps. SVG was adopted by Adobe, and is integrated in Firefox and Opera. There are a number of conversion tools to convert autocad DXF / DWG files, or ESRI Shape files to SVG files.

Object management

On top of the SVG map we've implemented an object layer for management of the objects. These objects will be maintained in a separate object file (2l dataset), related to the name of the SVG map. Each object (shown as an open or solid circle) has its own unique object id, and its xy coordinate position on the map.

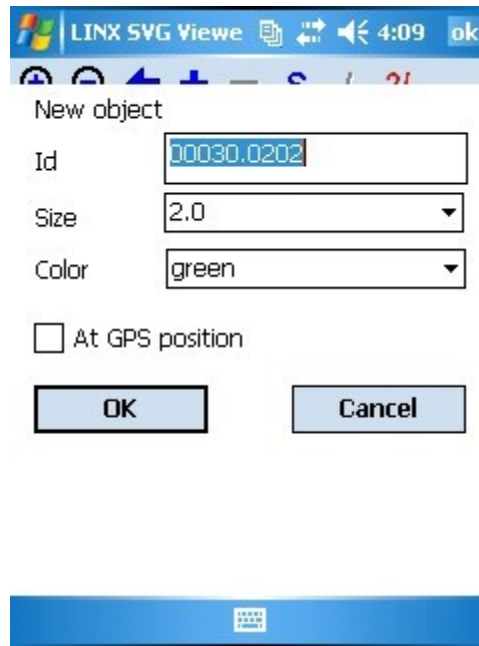


Add an object to the map

To add a new object to the map, select 'Add object' from the Edit menu (or press the large “+” icon in the toolbar).

Now fill in the next input screen

- Object Id, must be unique
- Object Size, select from the list
- Object color, select from the list



Finally click on the map at the location where the new object should be placed.

Note the program will already create a new unique id for you, based on the previous id filled in. If, for instance, the last object added to this map was lx00025, the new id will be lx00026. If you select a different color for the new object, this new color will be the default for the next new object.

If you fill in a non-unique object id, the program will give you an error message, and will not add a new object.

Select an object on the map

Start the select object function (either click the “S” icon on the toolbar or select Edit, Select object from the menu). Click on the object you want to select.

The newly selected object will be positioned in the center of the screen, the object will now be shown as a solid circle with a red border (or black border, in case the objects color was red). Selecting an object will change the selection date/time stamp in the objects record.

The previously selected object will loose its selection coloring (red/black border) and will be shown again as before it was selected.

Another way to select an object on the map, is double-click on the object to be selected.

Note: double-click on the selected object will start the inspection procedure for this object.

Delete an object from the map.

First you have to select the object you want to delete (see Select an object).

Next, either click the “-” icon on the toolbar, or select Edit, Delete object from the menu.

You will be prompted to confirm the delete action.

Note deleted objects will NOT be removed from the map, but will be shown as grayed circles. The object id will be prefixed with _X_ and a delete date/time stamp will be added to the object record.

Deleted objects cannot be changed or used for inspection, they can however be reselected and undeleted again.

Move an object to another location.

If the objects position on the map is incorrect (the object was added on the wrong spot or moved to an incorrect position), its position can be corrected with the move object function.

First select the object to be moved.

Now select Edit, Move object from the menu and click on the new position.

The object will be moved to this new position.

Moving an object will change the mutation date/time stamp in the object record.

Change an object id, size or color.

First select the object to be changed.

Now select Edit, Change object from the menu.

Fill in the proper id and/or change the size or color for this object.

The input screen to make the actual changes is similar to the one used to add a new object.

Changing an object will change the mutation date/time stamp in the object record.

Note 1, deleted objects cannot be changed, you first have to undelete them

Note 2, when you change the object id into an existing id, the change will not be accepted.

Locate an object using its object id

If you want to locate an object on the map, based on its unique object id, select Edit, Find object from the menu. Fill in the id you want to locate. If the object was found, it will automatically be selected and positioned in the center of the screen.

Locate your position on the map using GPS.

Select Options, GPS from the menu. Now the map will be centered on the actual GPS coordinates. A blinking GPS circle will be shown. Its color is an indication for the GPS accuracy, based on the number of satellites used to calculate the position:

Black : less than 3 GPS satellites, try to find another spot where more satellites can be seen.

Red : only 3 satellites were used to calculate the position (very inaccurate)

Orange : 4 or 5 satellites were used

Yellow: 6 or 7 satellites were used

Blue: 8 or more satellites were used

Inspect objects with 2L

First select the object you want to inspect. Now select "I" from the toolbar of Options, Switch to form from the menu or double-click on the selected item.

Inspection will change the inspection date/time stamp in the object record.

Inspected objects will be shown with a solid fill color, as compared to not-inspected objects, which

are shown as open circles.

The 2L inspection form (from which the map was started) will be displayed again.

The register variable SVGOBJECT will hold the ID of the selected object.

This ID can now be used to select the proper inspection-form and record to start the inspection on this object.

Selecting multiple objects

It is also possible to select more than one object and start inspecting the selected items.

First activate 'Select Multiple objects from the menu.

From now on each object that was not yet in the list of selected objects will automatically be added (shown by a light blue fill color). If the object was already in the list, it will be removed from the list again (showing its normal coloring again).

Note you cannot add a deleted object to your selection. The program will give you a warning message when you try.

If you want to add all the visible objects to the selection, use the menu function Select, Objects in view. All the visible objects that were not yet selected (and not deleted) will be added to the selection.

In the same way you can add objects from any rectangular area you select, using the menu function Select, Objects in rect.. and draw your own rectangular area.

When the selection you've made is incorrect, you can unselect individual objects, objects in an area, objects in the present view or your whole selection. See menu function Select, Unselect options.

Create line objects

A line object consists of 2 connected points

Select the menu function Options, Line.

Fill in the input screen for line-id, size and color

Press at the starting point position for your line on the map. This point will get <line-id>.1

Press at the endpoint position for the new line on the map. This point will get <line-id>.2.

Note: when you want to move the line, you have to select the start or endpoint individually, and move that point to the right position.

Create plot objects

A plot object consists of 2 or more connected points forming a closed polygon.

Select the menu function Option, Plot.

Fill in the input screen for plot-id, size and color

Now press the corner points of your plot, and press the Enter key to close your plot.

Individual plot points will be identified by <plot-id>.1, <plot-id>.2 etc..

The area of a closed plot object can be calculated and will be shown, using the menu function

View, Selected item (provided you've selected a plot object).

Create plot points

Plot points are individual single objects related to a plot.

To start creating or adding plot points, you first have to select an existing plot-object (by selecting one of the plot points of the plot-object).

Next select the menu function Options, Plotpoints.

Each click on the map will add a plot-point (named as <plot-id>.p1, <plot-id>.p2 etc.).

Press Enter to stop adding plot points.

Create tracking objects

When GPS is on, you can log individual tracks and show them on the map.

Select from the menu Options, Tracking

Fill in the track-id, size and color. Individual tracking points will be numbered <track-id>.1 etc..

The first tracking point starts at the present GPS position.

As soon as your actual GPS position is more than 5 m distance from your previous point a new tracking point will be added to your track, connected with gray dotted lines.

Unselect Tracking from the menu to stop the track.

Interaction between 2L and the SVG Viewer

Start the SVG Viewer from a 2L Form

Because the SVG Viewer on the handheld was installed as the default program to run SVG-files, you can start the viewer with the command RUN and specify your map as parameter.

Suppose your map is called MYCITY.SVG, you could use the command

```
RUN MYCITY.SVG
```

If your SVG map is not in the project folder, you have to specify the full path name to the SVG map. Like

```
RUN \SDCARD\SVG\MYCITY.SVG
```

Change the map without exiting the SVG Viewer

If you want to display another map (e.g. CITY2.SVG), you could use the command

```
RUN CITY2.SVG
```

This command will try to start the SVG Viewer, which in its turn will try to load and display the map CITY2.SVG.

If, however, the SVG Viewer is still running (and is still using MYCITY.SVG), this command will NOT start a new SVG Viewer, and the new map will not be displayed. It will only cause the running SVG Viewer to show its loaded map again.

But the SVG Viewer will check the registry for a variable SVGFILE to see if it has to reload a

different map.

Combining the RUN command with the registry setting from your 2L form will always display the intended map.

In case of CITY2.SVG the 2 commands will look like

```
SAVETOREG SVGFILE CITY2.SVG  
RUN CITY2.SVG
```

Change the default object color for new objects

Using the registry variable SVGCOLOR you can change the default color for new objects added to the map. The command for changing the color to red :

```
SAVETOREG SVGCOLOR red
```

Valid color names are

black purple blue red green olive brown orange yellow

Automatic selection of an object on the map.

If you have an object selected in your form (e.g. with object id LX007) and this object is also available in the object file of the map, you can use the registry variable 2LOBJECT to tell the SVG Viewer to select this object on the map. For LX007 this command will be:

```
SAVETOREG 2LOBJECT LX007
```

If the object LX007 does exist, this object will be selected, and the map will be centered on the object.

Note: if GPS-mode was activated, and the actual GPS position is not close to the selected object, the map will be centered on the actual position again.

Using information about the actual selected object in your forms.

In the paragraph “Inspect objects with 2L” the register variable SVGOBJECT was introduced. The contents of this variable holds the id of the last selected object on the map.

The next example will deal with a full session of the interaction between a 2L Form and the SVG Viewer.

Suppose we want to locate the existing object LX1499 (which we suppose to be the actual value of an input field MYOBJECT) on map MYMAP.SVG. And in case we did select another object on the map (e.g. LX1500) we would like to start our inspection form with this new object.

In our start form (START.FRM) we need at least an input field (MYOBJECT), were we can enter the object id LX1499, and a button with a start-function CMD TOTHEMAP.CMD

This command file TOTHEMAP.CMD could look like

```
REM specify the (new) map name to load in the SVG Viewer  
SAVETOREG SVGFILE MYMAP.SVG  
REM specify the (new) object to select, the contents of the input field MYOBJECT  
SAVETOREG 2LOBJECT %MYOBJECT%  
REM activate the map  
RUN MYMAP.SVG  
REM Set register variable SVGOBJECT to present object id
```

```

SAVETOREG SVGOBJECT %MYOBJECT%
REM Create a timer to check every second if a new object was selected on the map
SETTIMER 1 1 CHECKNEWOBJECT.CMD
STARTTIMER 1
REM start this timer

```

The timer command file CHECKNEWOBJECT.CMD could look like

```

REM Set onfail to check new selected object
ONFAIL :DOINSPECTION
REM Check if object is still the same
IF %_REG_SVGOBJECT%=%MYOBJECT%
REM Do nothing (an empty line is needed here)

REM Continue if another object was selected
:DOINSPECTION
REM Stop the timer, see note
REM Save this new object in the file MYOBJECT
LOADFROMREG SVGOBJECT MYOBJECT
REM switch to inspection form, with inspection dataset, and newly selected item
FORM INSPECTION.FRM INSPECTION.DAT %MYOBJECT%

```

Note: you do not need to stop the timer, because the command where you switch to another form, will automatically stop and kill all user timers.

Using advanced information

Use the object datafile as list to select in your form.

In the above example we could use the the map objects as a list-file attached to MYOBJECT. Since these objects for the MYMAP.SVG map are contained in the datafile MYMAP.DAT. The index column of this datafile contains the object id. Using this datafile as list we can select each of the objects on the map.

Getting additional information of the selected item.

Suppose we want to get the xy coordinates of the selected item, and its color. We can access the object datafile just like any 2L dataset, and search for a record using the command GETSEARCHREC.

The object datafile consists of 8 columns as specified in the header:

	id#x#y#r#color#sel#mut#del#type	
column 0	id	unique object id
column 1	x	x-coordinate
column 2	y	y-coordinate
column 3	r	radius of object, default 2.0 m
column 4	color	color of object
column 5	sel	date/time when object was last inspected
column 6	mut	date/time when object was last changed (moved / color changed)
column 7	del	date/time when object was marked as deleted
column 8	type	object type, empty for single objects, ln for line objects, pl for plot objects, tr for tracking objects

So getting info from object LX1500 and displaying its x / y position and color could be implemented with the following commands:

```
REM Fail if object not found
ONFAIL :NOTFOUND
REM Get object with GETSEARCHREC
GETSEARCHREC 0 MYMAP.DAT LX1500
REM Show coordinates + color
ALERT x=%SEARCHREC.1%, y=%SEARCHREC.2%, color=%SEARCHREC.4%
REM Empty line to finish command file

REM Continue if not found
:NOTFOUND
ALERT Object LX1500 not found!!!
```

Use color or part of the object code to specify the kind of object.

Suppose you have to inspect both lampposts and traffic lights in the same project with the same map.

You could use lp codes for lampposts (lp0023) with color yellow and tl codes for traffic lights (tl077) with color blue.

When you select lp0023 on the map and got this value from the SVGOBJECT registry variable into the MYOBJECT field, you could get the object type information (in this case the first 2 characters) in save these in your MYOBJECTTYPE field with a command like

```
SETMIDSTR MYOBJECTTYPE 1 2 %MYOBJECT%
```

Now if you have inspection forms specific for the different object types, you could switch to the correct inspection form with

```
FORM %MYOBJECTTYPE%INSPECTION.FRM %MYOBJECTTYPE
%INSPECTION.DAT %MYOBJECT%
```

Note: the proper inspection form (LPINSPECTION.FRM) must be present. Since file names under Windows based operating systems are not case sensitive, the objects lp0023, Lp55 and LP609 will all be inspected using this LPINSPECTION.FRM.