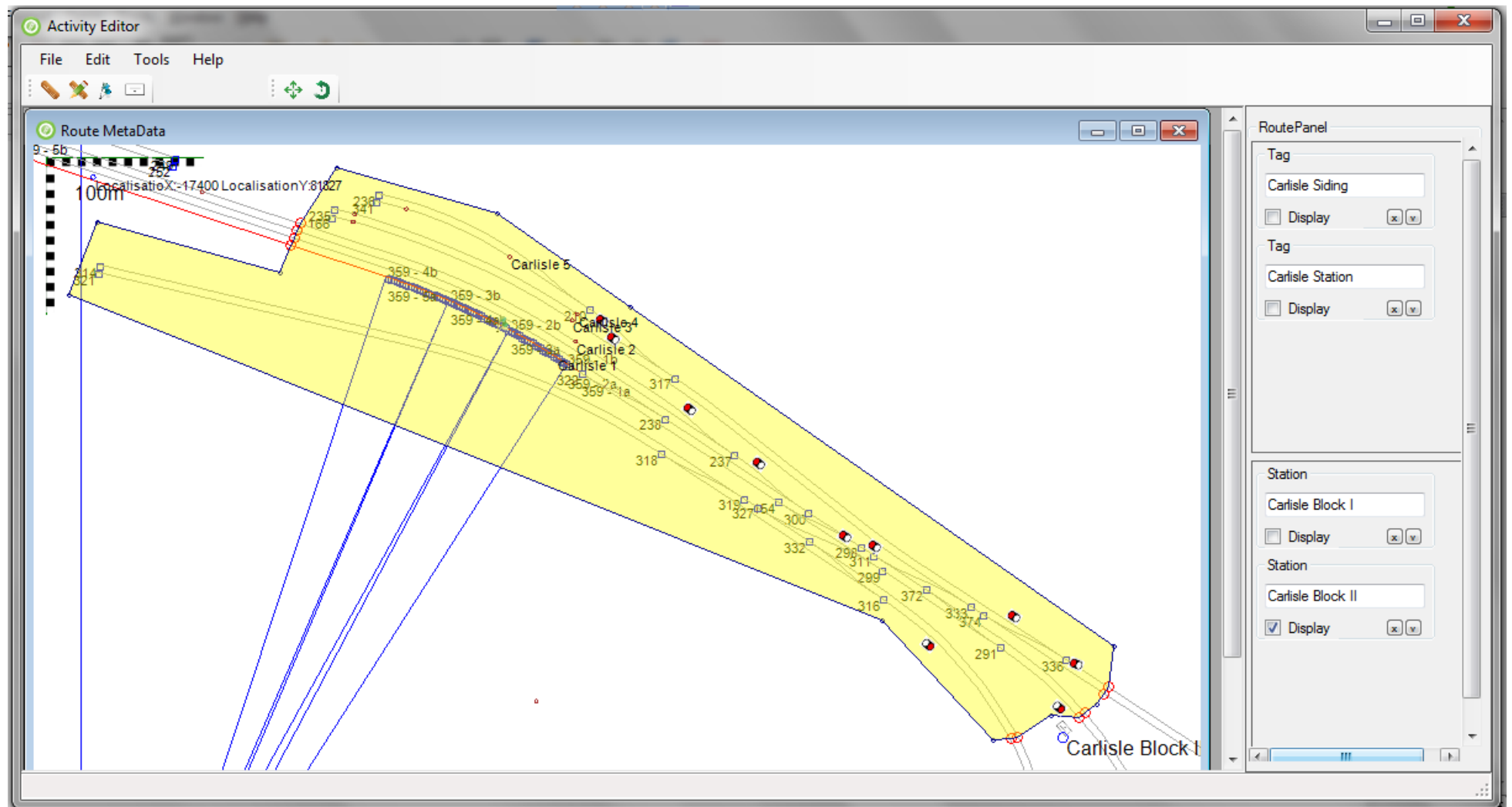


Open Rail

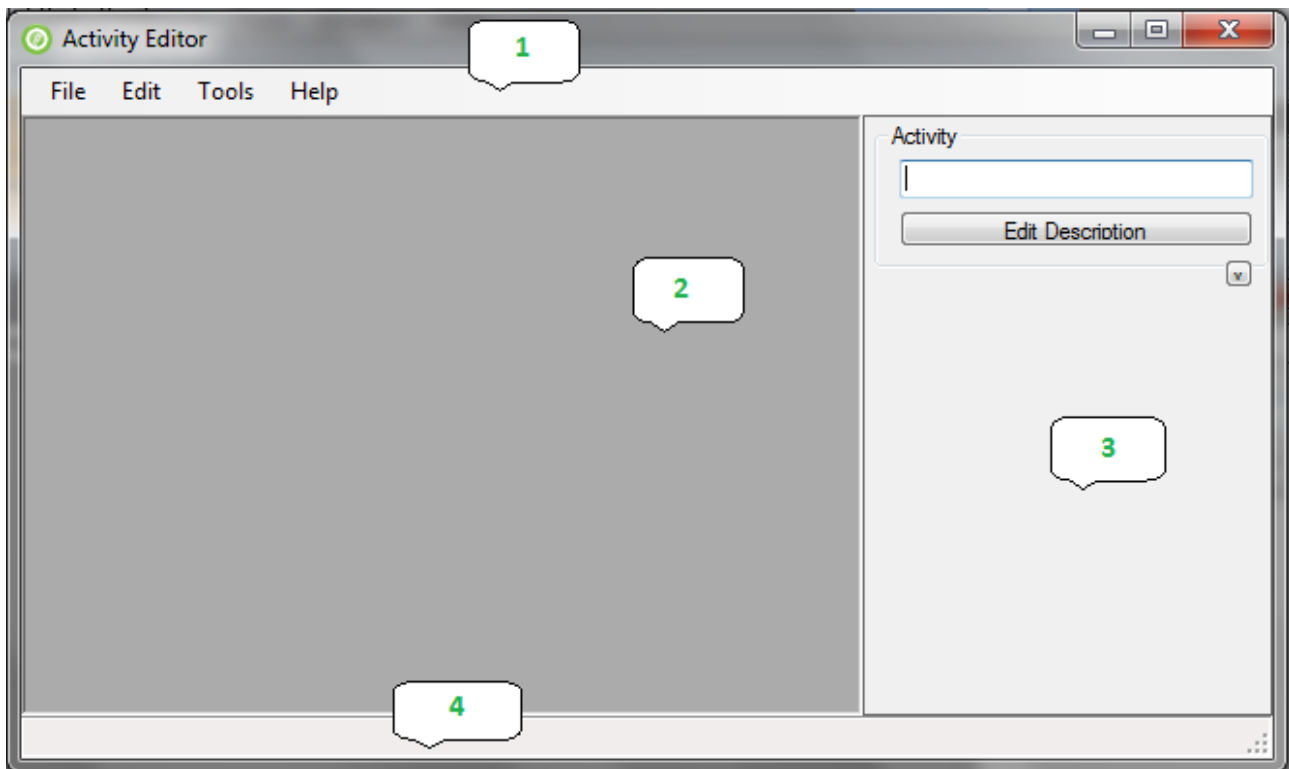
Activity Editor : current status

Here are some infos about what is available and the expected behaviour.



Carlisle with the Block II area. The platform, the switch number and the 'snap' sections are shown. All these informations can be turned off through the options.

The overall layout



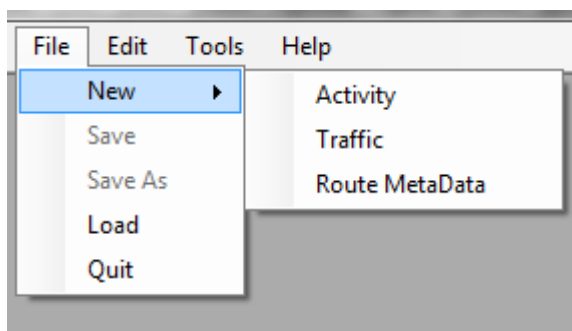
Not yet completely fixed but it's a start. 4 area are used :

1. The menu area:
2. The 'Layout' area :
3. The 'Information' area :
4. The status bar

The 'menu'

There are 4 menus for now.

File:



New Activity : bring the 'New Activity Wizard' in order to fix some of the activity parameters (Route used, name, ...)

New Traffic : To build generic traffic activities that can be used by normal activity

New Route MetaData : To give and complete specific datas that can be used by activities or OR

Save & Save As : To save the current item

Load : To load an existing item

Quit

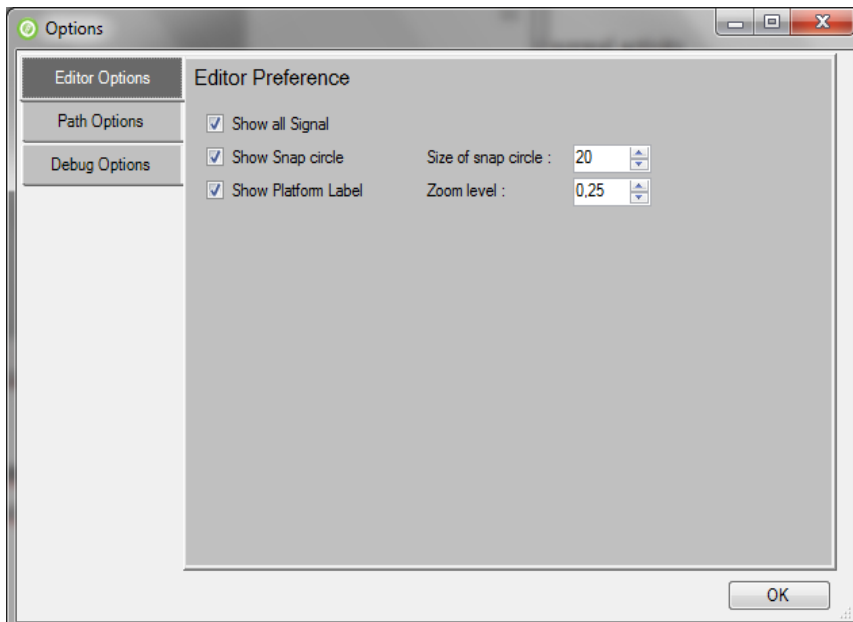
Edit :

Will be used to access some forms in order to manage the related information

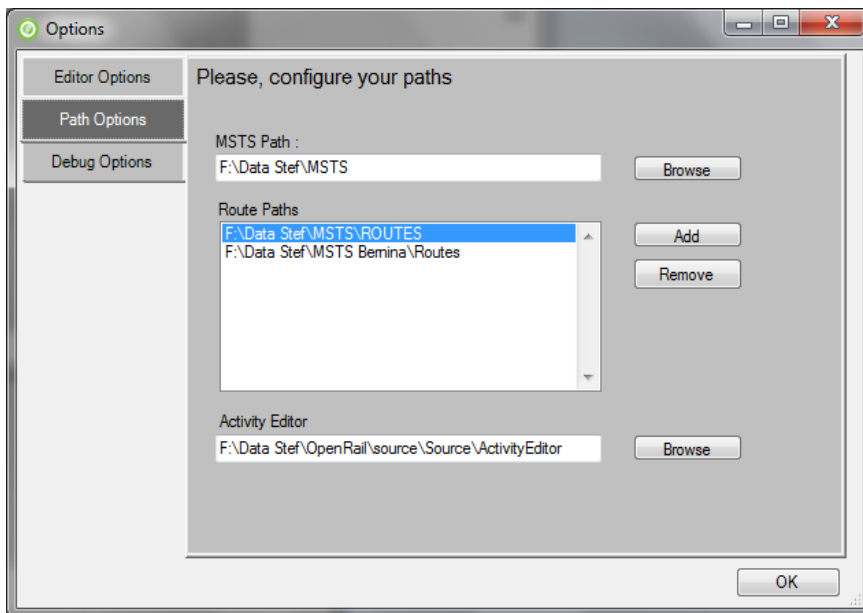
Tools :

Mainly used now to access the 'options' forms. The options are, for now, divided in three part :

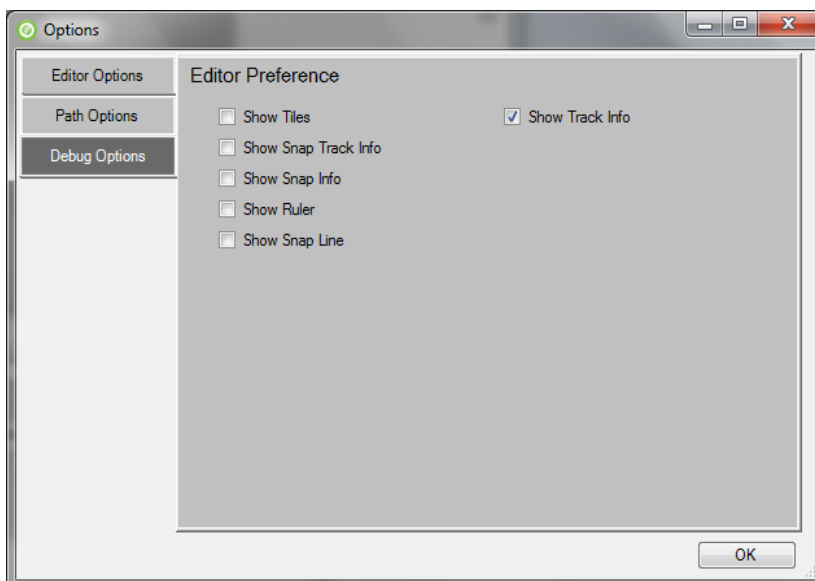
The 'Editor Preference',



The Path Options, used to define some important path. I think that I have to add some 'path option' for the saved data !

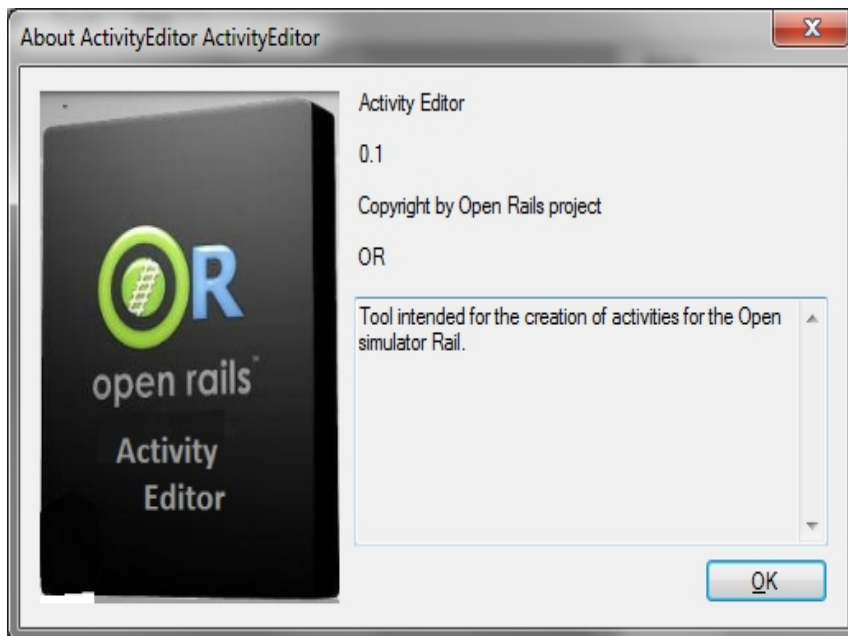


Debug Options , used to check, validate or complete information during development. May be kept, or not...



Help

Very simple at this stage, and to be defined



The Information Area

The screenshot shows a vertical panel titled "RoutePanel" containing two sections: "Tag" and "Station". Each section lists items with an editable text field, a "Display" checkbox, and two small buttons labeled "x" and "y".

Section	Item Name	Display	Buttons
Tag	Carlisle Siding	<input type="checkbox"/>	x y
	Carlisle Station	<input type="checkbox"/>	x y
	Cumwhinton	<input type="checkbox"/>	x y
Station	Carlisle Block I	<input type="checkbox"/>	x y
	Carlisle Block II	<input checked="" type="checkbox"/>	x y
	Carlisle Garage Block I	<input type="checkbox"/>	x y

It is used to display informations about tags and stations.

Here, we can see a list with the editable name, a CheckBox to display / not display the name, a small button to focus on the station and a button to remove it.

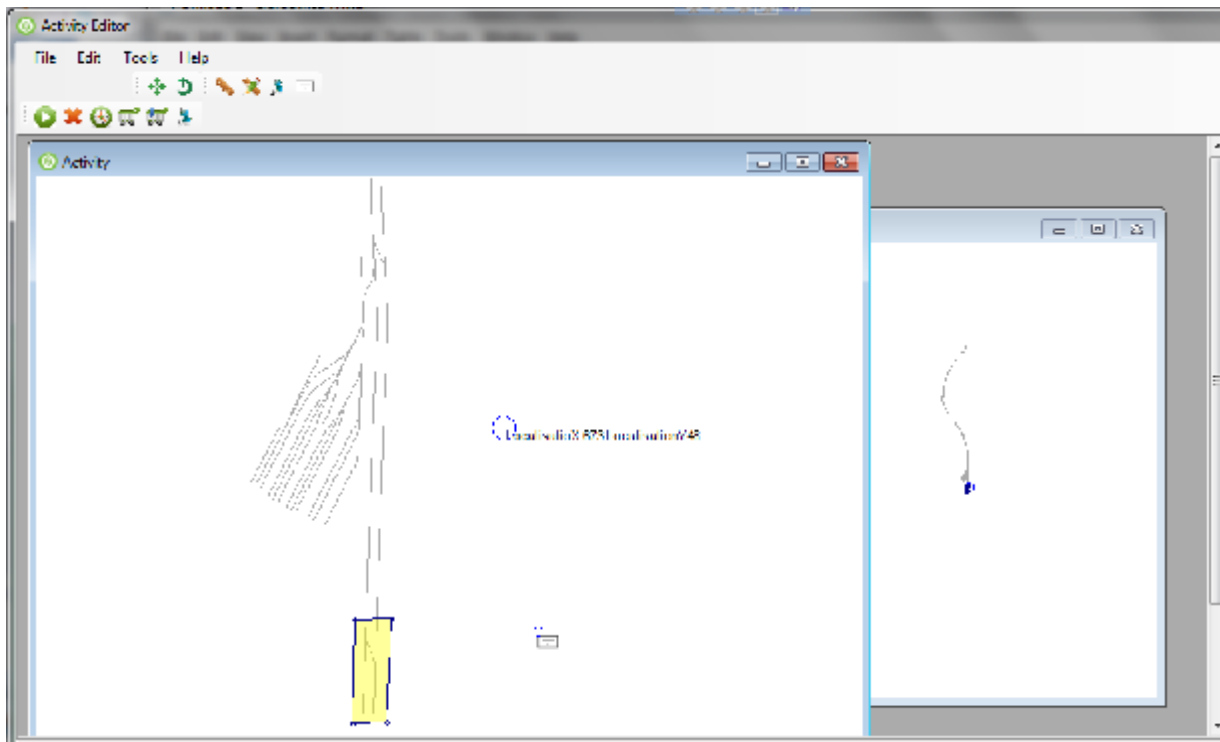
Needs a code review.

The Status Bar

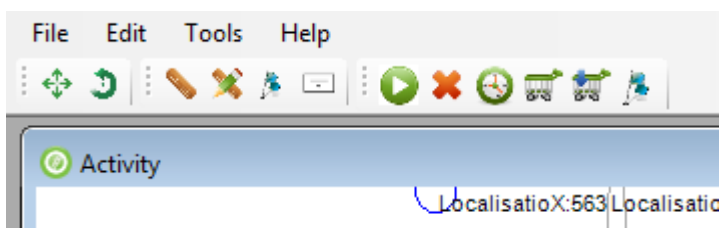
Gives information about what is done. Like when loading a route, or saving information, etc.

The Layout Area

We can load / create as many activities / route Metadata that we need. Each one is shown in a specific window in this area but there are no link between datas from each windows (may be we cans do this...).



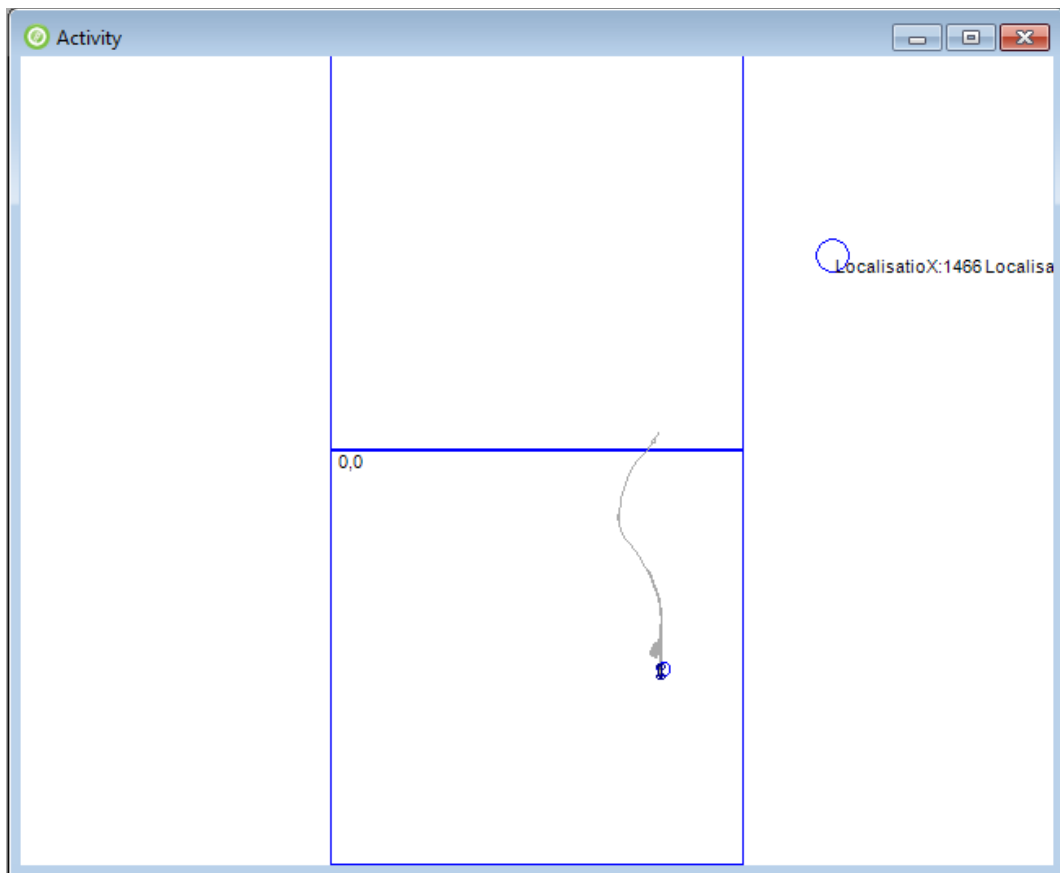
When editing Activity or Route Metadata, a set of specific tools are available under the menu.



Don't look at the current icon, they need to be redrawn. The tools are, from left to right :
Move, Rotate, Add or edit Area, edit area (will be removed, no more used), Tag, Station
Add Start, add Stop (will be removed, no more used), add Waiting point, add Action, add Reaction

What is possible inside a 'route layout' ?

When the program start for activity or Route Metadata editing, it shows all the route on the frame.

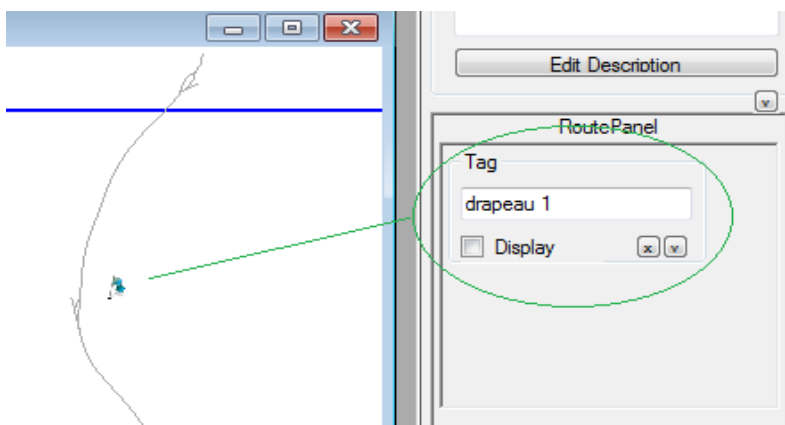


The program uses the tiles to compute the needed scale in order to display all the route. Than, we can zoom/unzoom and drag to see the details. I have to check the 'refreshing mode, some trouble.

Zoom/unzoom : by right click and mouse up (zoom) or mouse down (unzoom)

Drag : left click and move the mouse. When it reach the border, the windows continue to move

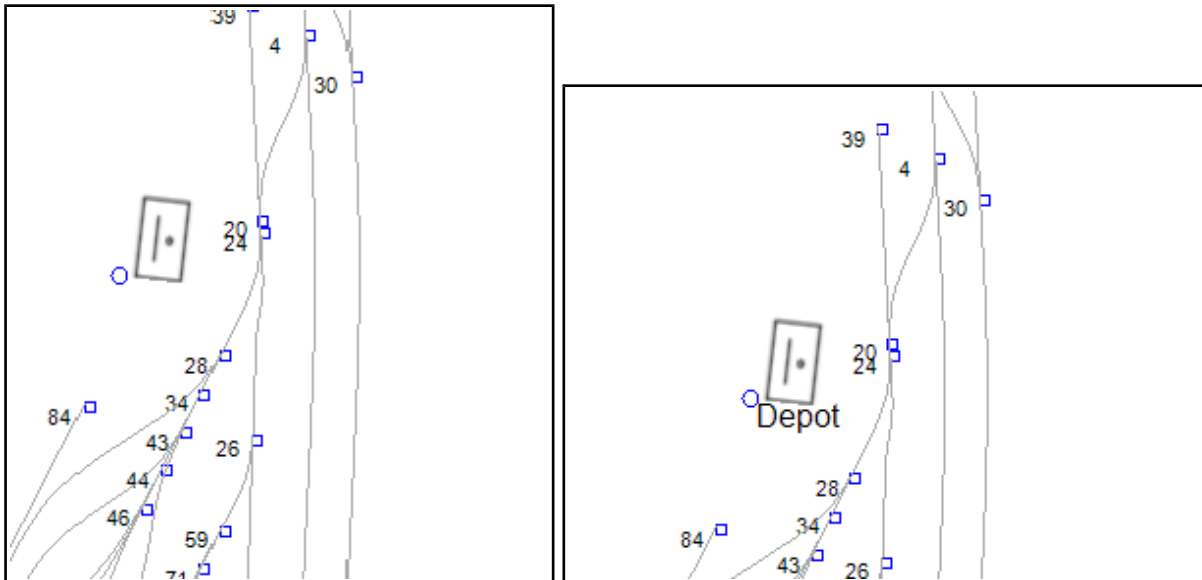
There is also the possibility to focus on tag or station, like shortcuts



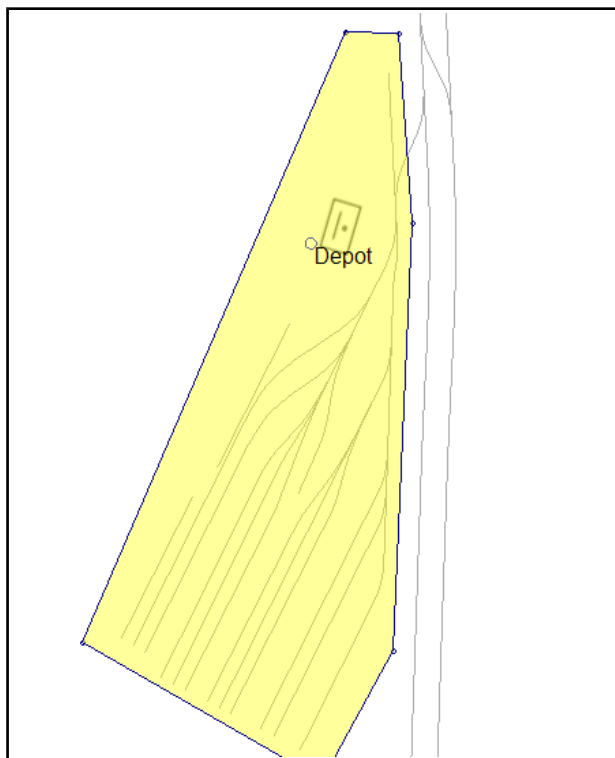
The 'route Metadata'

The goal : gives a way to define some 'area' on the route with attributes that can be used to ease the path design. But these informations can also be used by OR for other purpose. I think that in a future, the Route Editor can provide these facilities.

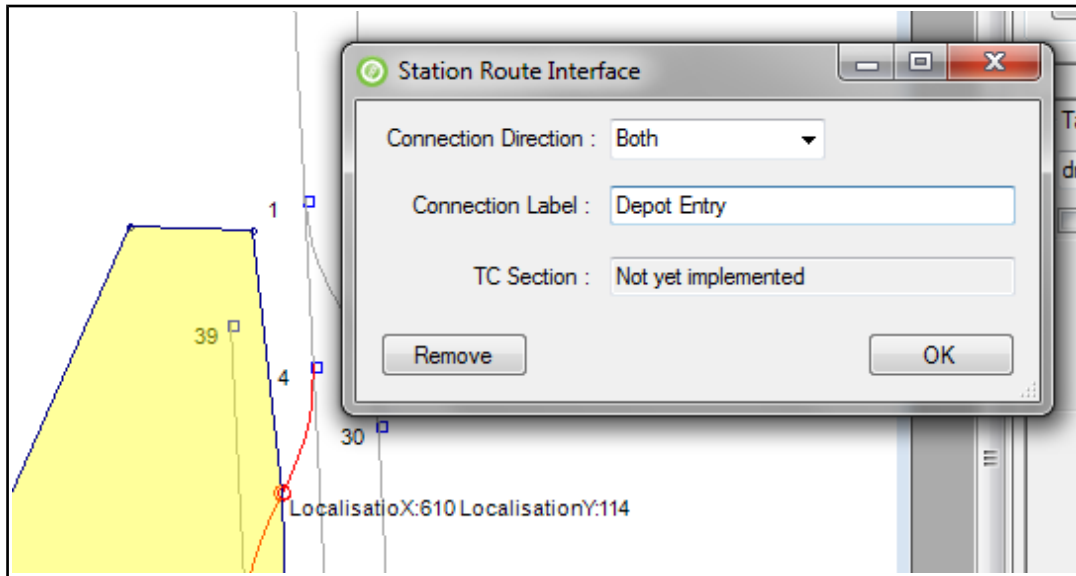
These areas are first shown as a small icon that represent a SignalBox. We can drag and rotate this icon, give it a name



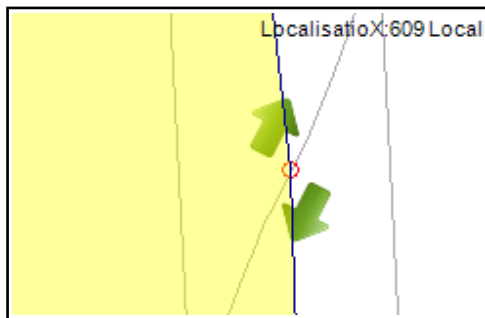
and add the yellow polygon that define the area.



At this stage, there is no connection with the track. It's just an information without interest except the shortcut to zoom on. But we can add some 'Connection Points' with attributes like 'the name' and the 'direction' : In, Out, Both.



The available directions are shown with arrows



At this stage, that's all.

The 'Activity'

That's the one on which I work now. I have to choose a way to define the activity informations, like :

The description : given with the Activity Wizard

The consist : given with the Activity Wizard

The path : to be done

The traffic : to be done

etc.

My intention is to define the path at first as a series of PathEvent like : Start, Stop, Wait, Revert, Action, Information, Signal, Switch, etc.

All these event inherit from a 'PathEventItem' which inherit from an 'ItemWidget'. The basic operation are defined in the 'ItemWidget' like 'move', 'edit', 'rotate'. The PathEventItem has the ability to 'stick / snap' to a track.

When we add them, a form with specific Activity datas will be shown :

The screenshot shows a window titled 'StartActivity'. It has two tabs: 'Activity Description' and 'Traffic Description'. The 'Activity Description' tab is active. Inside this tab, there is a section titled 'Activity Description' with an 'OK' button. Below this, there is a 'Description' section with 'Activity Name' (start244475) and 'Description' (empty text area). There is also a 'Consist' section (empty text area). To the right of the 'Description' section is a 'Path Events' section with 'Edit' and 'Remove' buttons.

It contains at least one tab with the specific Activity Infos

The 'Description' part will summarize the activity, the 'Path Events' part will shown all the events on current path.

And a possible list of 'Traffic Tab' with a similar layout, one by traffic. I also envisage to add one with information on the environment: Weather, external events,...

Activity Editor : To be done

Currently, I'm working on the activity tools where I have to finalize a 'way of' .

For the Route Metadata, I have to define a tool to compute all the available paths between a connector and : another connector, a buffer. On each sub path, the tool will add a list with all encountered platforms, a list with all encountered sidings. These informations will be used for Path Definition :

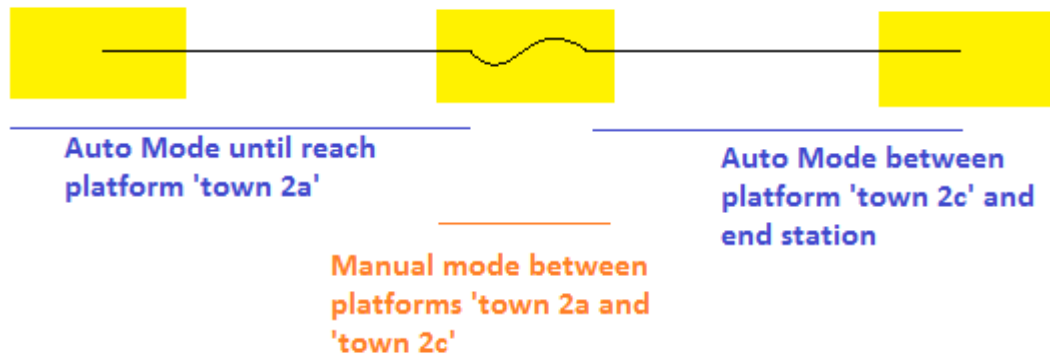
Create a Path like in MSTS, without any help from Route Metadata

Create a Path that join 'Station Area' with platform, set all the intermediate stations. The tools / runactivity can then decide which platform / siding to use.

Create a Path with a combination of the two others.

I start here, go to this station / platform, this station by using alternate path, ..., ending here.

One possibility is to define a path as a collection of paths between stations where the train must follow the rules and some movement inside a station in 'manual mode'.



Data exchange

The 'Route Metadatas' and the 'Activity datas' are stored in json files. The Editor Preferences are still in xml format. There is a library / files that can read / write these files to give OO interface.

No current test to use it with the 'RunActivity' program, it's a ToDo.

```
{
  "$id": "1",
  "$type": "LibAE.ORRRouteConfig, LibAE",
  "FileName": "Crich.cfg.json",
  "RoutePath": "F:\\Data Stef\\MSTS\\ROUTES\\Crich",
  "ItemWidget": {
    "$type": "System.Collections.Generic.List`1[[LibAE.ItemWidget, LibAE]], mscorlib",
    "$values": [
      {
        "$id": "2",
        "$type": "LibAE.StationWidget, LibAE",
        "nameStation": "station303984",
        "nameVisible": false,
        "stationArea": {
          "$type": "System.Collections.Generic.List`1[[LibAE.StationAreaWidget, LibAE]], mscorlib",
          "$values": [
            {
              "$id": "3",
              "$type": "LibAE.StationAreaWidget, LibAE",
              "Connector": null,
              "Location": {
                "$id": "4",
                "$type": "System.Drawing.PointF, System.Drawing",
                "IsEmpty": false,
                "X": 600.2367,
                "Y": -33.1972046
              },
              "Location2D": {
                "$id": "5",
                "$type": "System.Drawing.PointF, System.Drawing",
                "IsEmpty": false,
                "X": "-Infinity",
                "Y": "-Infinity"
              },
              "typeWidget": 0,
              "CoordMSTS": {
                "$id": "6",
                "$type": "LibAE.MSTSCoord, LibAE",
                "TileX": 0.0,
                "TileY": 0.0,
                "X": 600.2367,
                "Y": -33.1972046
              }
            }
          ]
        }
      }
    ]
  },
  ...
}
```