SpatiaLite v.2.4.0 – Release Candidate 2

Routing facility Addendum

Starting from v.2.4.0-RC2 the SpatiaLite's Routing facility [VirtualNetwork] is changed, because two different Shortest Path algorithms are now supported.

You can choose between:

- the classic **Dijkstra**'s algorithm [already implemented in previous versions]
- the A* algorithm [this algorithm applies *heuristic* assumptions, and in many cases may be noticeably faster then Dijkstra]

Important notice: you can still continue safely using any Network generated by previous SpatiaLite's version.

Anyway, only the Dijksta's algorithm can be supported when using old-format Networks.

So, be aware: in order to use the A* algorithm you must rebuild from scratch any Network related struct using v.2.4.0-RC2.

Building a Network:

Base Table [graph] NodeFrom Column Node from Oewery Columns geometry_columns_auth idx_roads_geometry node_to name oneway_formto oneway_formto oneway_formto idx_roads_geometry_rowing idx_roads_net NodesAuxT idx_roads_net NodesAuxT oneway_formto oneway_formto oneway_tofrom oneway_tofrom oneway_tofrom idx_roads_net NodesAuxT idx_roads_net NodesAuxT idx_roads_net NodesAuxT 0 using Length as Cost oneway_tofrom oneway_tofrom	Build Network		
OK Cancel	Base Table [graph] geometry_columns geometry_columns_auth idx_roads_geometry_node idx_roads_geometry_pare idx_roads_geometry_rowik idx_roads_net NodesAuxT idx_roads_net NodesAuxT idx_roads_net NodesAuxT idx_roads_net NodesAuxT layer_params layer_statistics layer_table_layout pattern_bitmaps project_defs raster_pyramids roads_net roads_net roads_net roads_net_data	NodeFrom Column Node To Column node_to node_to name oneway_fromto oneway_fromto oneway_fromto oneway_fromto oneway_fromto Outri-Directional Outring Length as Cost Bi-Directional Outring Cost Column Supported Shortest Path Algorithms A* + Dijkstra Dijkstra only Oneway_formto oneway_formto Oneway_formto oneway_formto Oneway_formto oneway_formto oneway_tofrom oneway_form Iength oneway_tofrom Iength Iength OK Cancel	Geometry Column oneway_tofrom length cost geometry Cost Column oneway_tofrom length cost geometry Cost Column Name Column Mame Column Mame Column Mame Column Mame Column Name Column Col

While creating a Network using v.2.4.0-RC2 you can:

- support the Dijkstra's algorithm only
 - $^{\circ}\,$ this option produces a VirtualNetwork fully compatible with previous SpatiaLite's versions
- support both algorithms [Dijkstra's and A*]
 - using this option requires a small extra-space: not so much, anyway

The same option is available for the CLI tool **spatialite_network** as well.

Querying a VirtualNetwork table:

Submitting a Routing request is exactly the same as in previous versions:

```
SELECT *
FROM roads_net
WHERE NodeFrom = 9200418 AND NodeTo = 251369545
```

But using v.2.4.0-RC2 the returned resultset is slightly different:

Algorithm	ArcRowid	NodeFrom	NodeTo	Cost	Geometry
Dijkstra	NULL	9200418	251369545	17.939336	LINESTRING
Dijkstra	3	9200418	277357928	6.402357	NULL
Dijkstra	4	277357928	260914932	3.568285	NULL
Dijkstra	5	260914932	251369545	7.968694	NULL

As you can notice, a new **Algorithm** column is now returned [corresponding to the algorithm actually used for the current routing solution]

<u>Please note</u>: by default the Dijksta's algorithm is the currently selected one.

You can switch from one algorithm to the other simply executing:

UPDATE roads_net SET Algorithm = 'A*'
or
UPDATE roads_net SET Algorithm = 'Dijkstra'

Please note: the above statement is effectless if the underlying VirtualNetwork doesn't actually supports the A* algorithm.

And you can check wich one algorithm is currently selected performing the following query: **SELECT** algorithm **FROM** roads net

That's all, folks