

SQL Cluster and Subcluster

Version 6.0.0 has introduced a feature where the results of a standard SQL statement are scanned for field names which should be interpreted as a cluster or subcluster.

No assumption has been made that the values 'cluster' and 'subcluster' would translate into other languages, and I also wanted to make it possible for someone to be able to query a column named 'cluster' in their data (as you might find in an infrastructure workbook). Therefore, I have added new SQL settings on the 'settings' tab:

Command Options	'Graphviz' Tab	'Source' Tab	'data' Worksheet	'sql' Worksheet															
Graph Options	'Extensions' Tab	'Exchange' Tab	'source' Worksheet	'styles' Worksheet															
<div>Comment/Error Flag Column: <input type="text" value="A"/></div> <div>SQL Statement Column: <input type="text" value="B"/></div> <div>Excel File Column: <input type="text" value="C"/></div> <div>Status Column: <input type="text" value="D"/></div> <div>Filter Column (Ribbon): <input type="text"/></div> <div>Filter Value (Ribbon): <input type="text"/></div> <div><table><thead><tr><th></th><th>Cluster Field Names</th><th>Subcluster Field Names</th></tr></thead><tbody><tr><td>Label:</td><td><input type="text" value="CLUSTER"/></td><td><input type="text" value="SUBCLUSTER"/></td></tr><tr><td>Style Name:</td><td><input type="text" value="CLUSTER STYLE NAME"/></td><td><input type="text" value="SUBCLUSTER STYLE NAME"/></td></tr><tr><td>Attributes:</td><td><input type="text" value="CLUSTER ATTRIBUTES"/></td><td><input type="text" value="SUBCLUSTER ATTRIBUTES"/></td></tr><tr><td>Tooltip:</td><td><input type="text" value="CLUSTER TOOLTIP"/></td><td><input type="text" value="SUBCLUSTER TOOLTIP"/></td></tr></tbody></table><div>Label Split Field Names</div><div>Split Length: <input type="text" value="SPLIT LENGTH"/></div><div>Line Ending: <input type="text" value="LINE ENDING"/></div><div>Count Substitution Strings</div><div>Cluster: <input type="text" value="%clc%"/></div><div>Subcluster: <input type="text" value="%scc%"/></div><div>Record Set: <input type="text" value="%rsc%"/></div></div>						Cluster Field Names	Subcluster Field Names	Label:	<input type="text" value="CLUSTER"/>	<input type="text" value="SUBCLUSTER"/>	Style Name:	<input type="text" value="CLUSTER STYLE NAME"/>	<input type="text" value="SUBCLUSTER STYLE NAME"/>	Attributes:	<input type="text" value="CLUSTER ATTRIBUTES"/>	<input type="text" value="SUBCLUSTER ATTRIBUTES"/>	Tooltip:	<input type="text" value="CLUSTER TOOLTIP"/>	<input type="text" value="SUBCLUSTER TOOLTIP"/>
	Cluster Field Names	Subcluster Field Names																	
Label:	<input type="text" value="CLUSTER"/>	<input type="text" value="SUBCLUSTER"/>																	
Style Name:	<input type="text" value="CLUSTER STYLE NAME"/>	<input type="text" value="SUBCLUSTER STYLE NAME"/>																	
Attributes:	<input type="text" value="CLUSTER ATTRIBUTES"/>	<input type="text" value="SUBCLUSTER ATTRIBUTES"/>																	
Tooltip:	<input type="text" value="CLUSTER TOOLTIP"/>	<input type="text" value="SUBCLUSTER TOOLTIP"/>																	

Clusters and subclusters can specify:

- The cluster column, where its value is placed in the 'label' column.
- A Style Name, which should map to a style on the 'styles' worksheet. This value can be a static string, a composite string using substitution and/or concatenation, or a column value which matches a style name on the 'styles' worksheet. An example of the last scenario is when creating heat maps where you might have values such as 'Critical' and 'Standard', and then have node styles named 'Critical' with a red colored fill, and 'Standard' with a blue colored fill.
- Attributes, which allow you to include line-specific values.
- Tooltip, which is useful for hover text in SVG output.

The values above are shown in all capitals; however, the implementation is case-insensitive, and CLUSTER, Cluster, ClUsTeR, and cluster would all be recognized as the same.

There are also substitution strings which for the count value as clusters, subclusters, and records are iterated over. Before data is transferred to the 'data' worksheet a 'replace' is performed against the substitution string with the current value of the count.

Command Options	'Graphviz' Tab	'Source' Tab	'data' Worksheet	'sql' Worksheet
Graph Options	'Extensions' Tab	'Exchange' Tab	'source' Worksheet	'styles' Worksheet

Comment/Error Flag Column:
SQL Statement Column:
Excel File Column:
Status Column:

Heading Row:
First Row:

Filter Column (Ribbon):
Filter Value (Ribbon):

Cluster Field Names
Label:
Style Name:
Attributes:
Tooltip:

Subcluster Field Names

Label Split Field Names
Split Length:
Line Ending:

Count Substitution Strings
Cluster:
Subcluster:
Record Set:

The counts are useful if you want to specify different border styles per cluster/subcluster or want to emit a sort order in a 'sortv' attribute. 'sortv' is very useful when using the osage layout to create a heatmap or domain model where you want to control the order of columns or elements within a column.

There is nothing special about the % values in the substitution strings; you may change them to anything you like. The assumption is to specify something which would be unlikely to appear in the data. The % harkens back to the C language, where %d represents a digit when converting to string.

Label Splitting

Another new capability allows you to split long strings into multi-line labels and designate the line ending to use to make the label left, center, or right justified.

Command Options	'Graphviz' Tab	'Source' Tab	'data' Worksheet	'sql' Worksheet
Graph Options	'Extensions' Tab	'Exchange' Tab	'source' Worksheet	'styles' Worksheet

Comment/Error Flag Column:	A	Heading Row:	1
SQL Statement Column:	B	First Row:	2
Excel File Column:	C		
Status Column:	D		
Filter Column (Ribbon):			
Filter Value (Ribbon):			

	Cluster Field Names	Subcluster Field Names
Label:	CLUSTER	SUBCLUSTER
Style Name:	CLUSTER STYLE NAME	SUBCLUSTER STYLE NAME
Attributes:	CLUSTER ATTRIBUTES	SUBCLUSTER ATTRIBUTES
Tooltip:	CLUSTER TOOLTIP	SUBCLUSTER TOOLTIP

	Label Split Field Names
Split Length:	SPLIT LENGTH
Line Ending:	LINE ENDING

	Count Substitution Strings
Cluster:	%clc%
Subcluster:	%scc%
Record Set:	%rsc%

To split labels add the fragment below to your SQL statement:

'5' as [SPLIT LENGTH], '\n' as [LINE ENDING],

In this example the label will be split into multiple lines as closely as possible to 5 characters. The splits occur where blanks are found, so any word exceeding 5 characters would be preserved at its full length.

Line endings can be any string. The most commonly used are:

- '\n' - New Line, center alignment
- '\r' - New Line, right alignment
- '\l' - New Line, left alignment
- '|' - Pipe delimited (useful for Record shapes)

Configurable Cluster Style Naming

Another change is that the cluster and subcluster style names will be modified at run-time to end with a begin/end suffix. The suffix values are also configurable, but appear on the 'styles' worksheet settings as the style designer has been updated to use them as well in place of the currently hard-coded values 'Begin' and 'End'.

Command Options	'Graphviz' Tab	'Source' Tab	'data' Worksheet	'sql' Worksheet
Graph Options	'Extensions' Tab	'Exchange' Tab	'source' Worksheet	'styles' Worksheet

Comment Column:	A	Heading Row:	1
Style Column:	B	First Row:	2
Format Column:	C	Last Row:	0
Object-Type Column:	D	subgraph-open Name Suffix:	Begin
First Yes/No View Column:	E	subgraph-close Name Suffix:	End

For example, if you specify in your SQL query a value such as:

```
'Border %scc% ' as [subcluster style name],
```

The 3rd subcluster will contain style names of

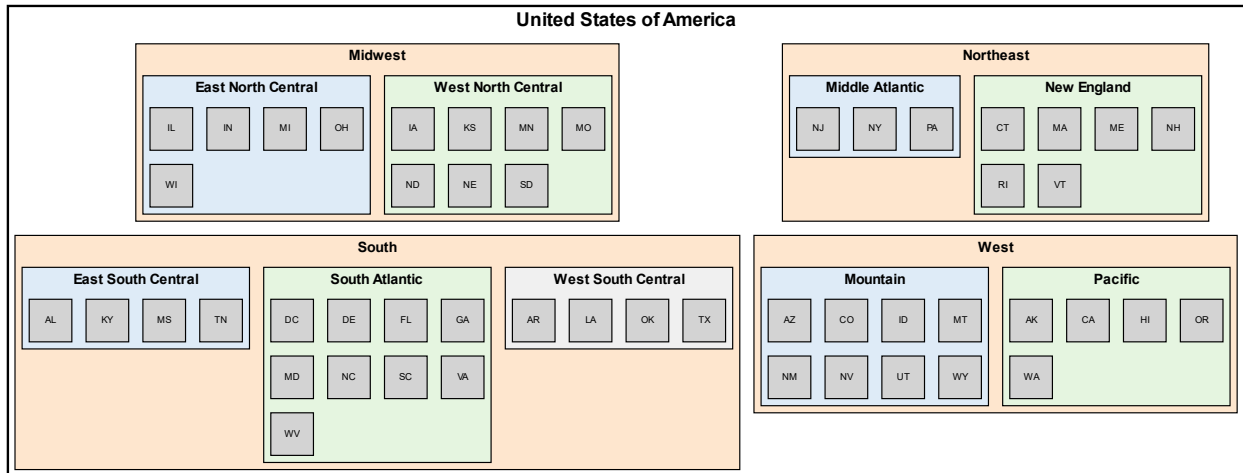
```
'Border 3 Begin'      on open, and
'Border 3 End'        on close.
```

Remember to include that trailing space if you are using the pre-defined borders on the 'styles' worksheet.

You can designate the clustering field names of your choice, then use them in your queries as follows:

```
SELECT
  [State Code]          as [Item],
  'Medium Square'       as [style name],
  'sortv=%rsc%'         as [attributes],
  [State Code]          as [Label],
  [State]               as [External Label],
  [State]               as [Tooltip],
  [Region]              as [cluster],
  'Border 4 '           as [cluster style name],
  [Region]              as [cluster tooltip],
  'sortv=%clc% packmode=array_utr3' as [cluster attributes],
  [Division]            as [subcluster],
  'Border %scc% '       as [subcluster style name],
  [Division]            as [subcluster tooltip],
  'sortv=%scc% packmode=array_utr4' as [subcluster attributes]
FROM
  [census regions$]
ORDER BY
  [Region]      ASC,
  [Division]    ASC,
  [State Code] ASC
```

This SQL query produces the Region and Division clusters within the “United States of America” cluster in this graph created with the osage layout.



Notice that the subclusters each change color due to the subcluster’s count being included in the subcluster style name.

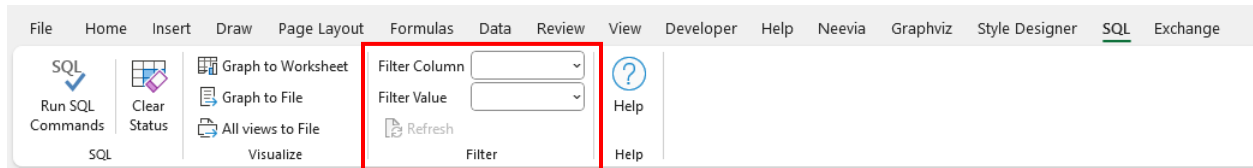
Note: I also added a 3rd view to the ‘styles’ worksheet named ‘No Edges’ which filters out the edge relationships making it easier to use the osage layout as intended.

Batch SQL

The SQL capabilities are very powerful, but in the past if you wanted to produce different graphs from the same data you needed to have one spreadsheet per set of SQL statements. Now it is possible to have a long list of statements and designate which ones to run using filtering techniques.

Filtering

Two new drop downs have been added on the SQL Tab.



The first lets you choose a Filter Column. Once you choose a column the Filter Value drop down is populated with the unique values in that column. From that list you choose a value, then click Run SQL. Only the SQL statements which have matching filter values get executed.

Auto Publishing

Additional commands have been added to the SQL column to trigger graph generation as the SQL statements are run.

The syntax rules for publishing are:

- RESET
- PREVIEW [AS DIRECTED | UNDIRECTED GRAPH]
- PUBLISH [ALL VIEWS] [AS DIRECTED | UNDIRECTED GRAPH] [file prefix]

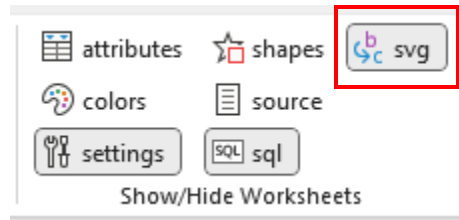
Where RESET clears the 'data' worksheet, PREVIEW runs the "Graph to Worksheet" button, PUBLISH runs the "Graph to File" button, and [file prefix] lets you specify the base value for the file name.

The valid combinations are:

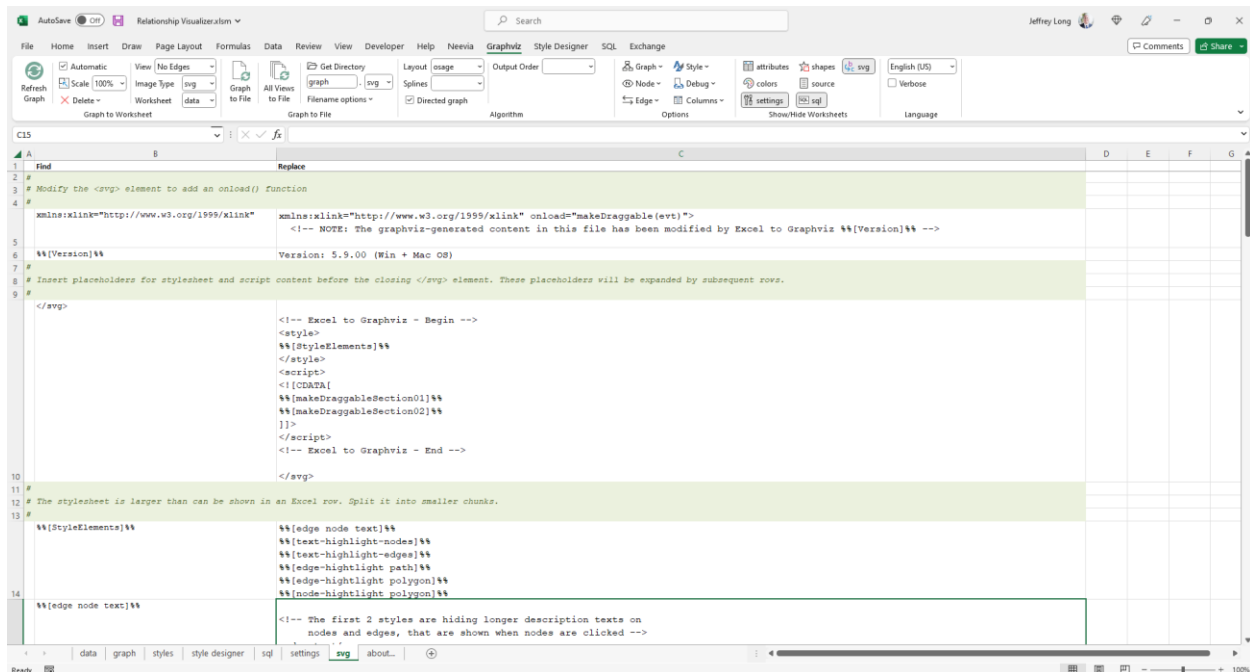
- PREVIEW
- PREVIEW AS DIRECTED GRAPH
- PREVIEW AS UNDIRECTED GRAPH
- PUBLISH
- PUBLISH prefix
- PUBLISH AS DIRECTED GRAPH prefix
- PUBLISH AS UNDIRECTED GRAPH prefix
- PUBLISH ALL VIEWS
- PUBLISH ALL VIEWS prefix
- PUBLISH ALL VIEWS AS DIRECTED GRAPH prefix
- PUBLISH ALL VIEWS AS UNDIRECTED GRAPH prefix

SVG Post-Processing

A capability to post-process SVG files created by the “Graph to File” button has been introduced. On the Graphviz tab you will see a new button to show/hide a ‘svg’ worksheet.



The new svg worksheet appears as follows:



It has 3 columns, the usual comment indicator column which designates the row should be skipped, a “find” string, and a “replace” string.

Excel can store a large block of text in a cell, but it will not display all of it or make it easy to scroll and edit. Text needs to be broken into chunks to be able to view them within a row. Some of the chunks contain placeholders to chunks which are specified further down the worksheet.

For example, the JavaScript needs to be inserted into the bottom of the SVG file, but before the `<svg/>` element, so the `<svg/>` element is replaced with:

```
<!-- Excel to Graphviz - Begin -->
<style>
%%[StyleElements]%%
</style>
<script>
```

```

<![CDATA[
%%[makeDraggableSection01]%%
%%[makeDraggableSection02]%%
]]>
</script>
<!-- Excel to Graphviz - End -->

</svg>

```

When row 14 is processed, `%%[StyleElements]%%` in the above fragment is replaced with:

```

%%[edge node text]%%
%%[text-highlight-nodes]%%
%%[text-highlight-edges]%%
%%[edge-highlight path]%%
%%[edge-highlight polygon]%%
%%[node-highlight polygon]%%

```

Converting it to:

```

<!-- Excel to Graphviz - Begin -->
<style>
%%[edge node text]%%
%%[text-highlight-nodes]%%
%%[text-highlight-edges]%%
%%[edge-highlight path]%%
%%[edge-highlight polygon]%%
%%[node-highlight polygon]%%
</style>
<script>
<![CDATA[
%%[makeDraggableSection01]%%
%%[makeDraggableSection02]%%
]]>
</script>
<!-- Excel to Graphviz - End -->

</svg>

```

As each row is processed the placeholders are replaced with the actual CSS style definitions or JavaScript code. It is a bit confusing at first, and admittedly a little bit kludgy but it gets around Excel's cell limitations for text data.

Note, there is nothing special about the `%%[...]%%` notation on the find strings. That pattern was used to ensure there would be no accidental substitutions of text.

Any value can be used in the find/replace, as illustrated on row 5 where:

```
xmlns:xlink=http://www.w3.org/1999/xlink"
```

is replaced with:


```
xmlns:xlink="http://www.w3.org/1999/xlink"  
onload="makeDraggable(evt)">
```

The SVG post-processing is only performed by the “Graph to File” and “All Views to File” buttons. Microsoft Office 365 allows you to display SVG images in the workbook, but the hover and Javascript options are not supported. I think Microsoft changes the SVG to a picture element when displaying SVG.

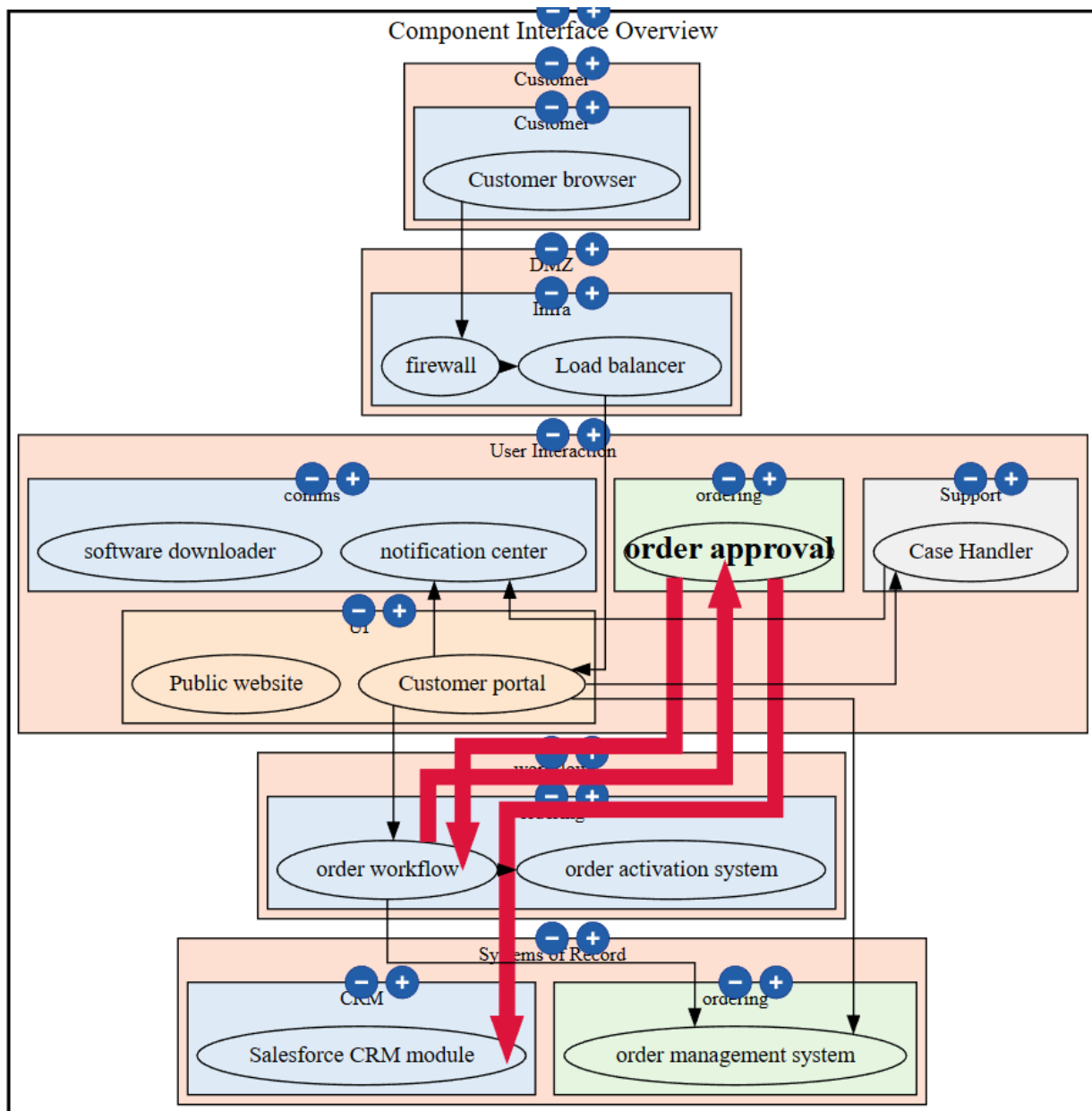
The post-processing logic can also be toggled on/off on the ‘SVG’ tab through the “Postprocess SVG Output” checkbox. This toggle will allow you to generate “pure” Graphviz output if desired.

The postprocessing feature is toggled OFF in the download file. You must toggle it on to use the feature.



The default implementation includes contributed Javascript which will let you click on a node or an edge and see the border change color and size. Edges are animated from tail to head, and zoom capabilities are provided for clusters.

In the following example the “order approval” node has been selected.

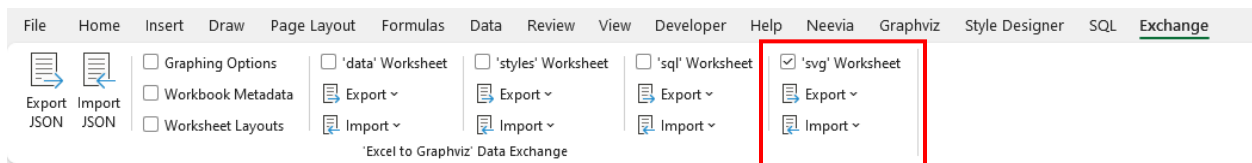


Import/Export

The enhancement for SQL clustering and SVG postprocessing has been incorporated into the Exchange tab. The new values on the 'settings' worksheet have been added to their respective sections in the JSON file on export, and will be pulled into the settings on Import.

The 'svg' worksheet has also been added to the JSON exchange file. The enhanced JSON format will continue to let the contents be placed under version control, or make it easy to move to future releases.

The Exchange tab now appears as:



Where you can see there is a new group for 'svg' Worksheet.

You can also use Exchange to just export/import the svg worksheet data, without all the other spreadsheet content. It would be possible to create different versions of the post-processing statements which could be swapped in/out. For example, if you wanted different css styles to change edge colors or thickness.