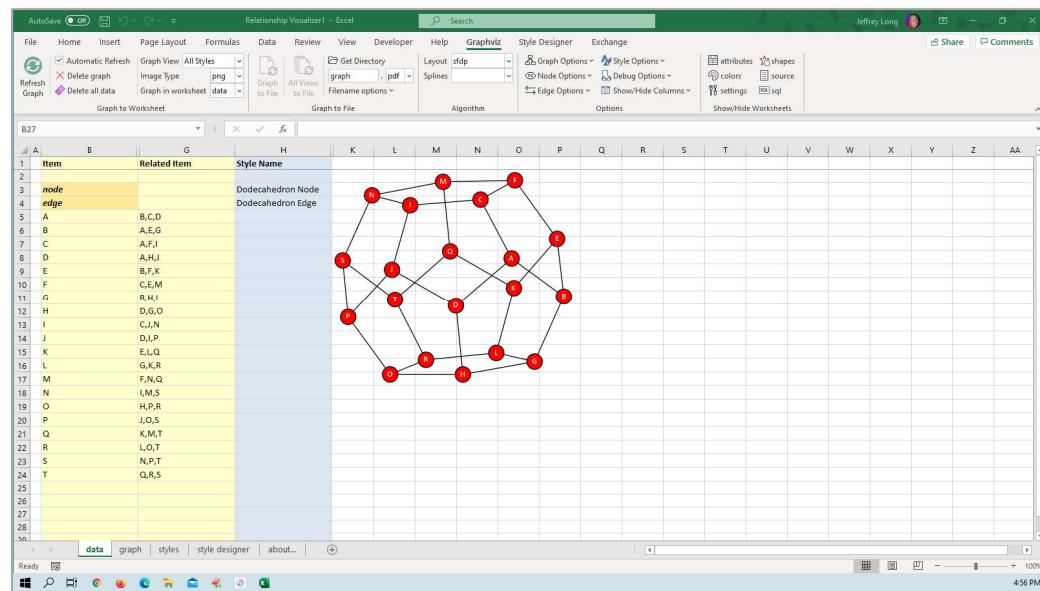


EXCEL TO GRAPHVIZ FOR WIN OS



4/10/2022

See the connections

The **Excel to Graphviz Relationship Visualizer** makes it easy for you to collect pairs of data in Excel and send the data through the open source Graphviz graphing engine to draw the data connections. The combination of these tools makes it quick and easy to draw complex data relationships.

Excel to Graphviz for Win OS

SEE THE CONNECTIONS

Excel to Graphviz Relationship Visualizer

Copyright © 2015-2022, Jeffrey J. Long. All rights reserved.

Written and published by Jeffrey J. Long

Published at: <https://sourceforge.net/projects/relationship-visualizer/>

Contact the author at Relationship.Visualizer@gmail.com

This program is free software; you can redistribute it and/or modify it under the terms of the [MIT License](#).

MIT LICENSE

Copyright (c) 2015-2022 Jeffrey Long

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Third-Party Notices / Author Acknowledgements

Excel to Graphviz Relationship Visualizer uses third-party libraries or other resources that may be distributed under licenses different than the one from this software. In the event that I accidentally failed to list a required notice, please bring it to my attention.

Graphviz

<http://www.graphviz.org>

Written By: [Graphviz Project Contributors](#)

Version: 3.0.0 (20220226.1711)

Date: 26 February 2022

License: Graphviz is free software licensed under the [Eclipse Public License \(EPL\) v1.0](#)

Shell and Wait

<http://www.cpearson.com/excel/ShellAndWait.aspx>

Written By: [Charles H. Pearson](#)

Version: unknown

Date: 06 November 2013

Copyright © 2018, Charles H. Pearson

License: All of the formulas and VBA code are explicitly granted to the Public Domain.

Brewer Color Schemes

Apache-Style Software License for ColorBrewer software and ColorBrewer Color Schemes, Version 1.1

Copyright (c) 2002 Cynthia Brewer, Mark Harrower, and The Pennsylvania State University. All rights reserved.

1. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
2. Redistributions as source code must retain the above copyright notice, this list of conditions and the following disclaimer.

The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

This product includes color specifications and designs developed by Cynthia Brewer (<http://colorbrewer.org/>).

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

3. The name "ColorBrewer" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact Cynthia Brewer at cbrewer@psu.edu.
4. Products derived from this software may not be called "ColorBrewer", nor may "ColorBrewer" appear in their name, without prior written permission of Cynthia Brewer.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL CYNTHIA BREWER, MARK HARROWER, OR THE PENNSYLVANIA STATE UNIVERSITY BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

VBA-JSON - Version: 2.3.1

<https://github.com/VBA-tools/VBA-JSON/>

Written By: [Tim Hall](#)

Copyright © 2016, Tim Hall

License: VBA-JSON is free software licensed under the [MIT License](#)

VBA-Dictionary - Version: 1.4.1

<https://github.com/VBA-tools/VBA-Dictionary>

Written By: [Tim Hall](#)

Copyright © 2016, Tim Hall

License: VBA-Dictionary is free software licensed under the [MIT License](#)

VBA Stopwatch

<https://github.com/hubisan/excel-vba-stopwatch.cls>

Written By: Daniel Hubman

Copyright © 2016, Daniel Hubmann

License: VBA-Dictionary is free software licensed under the [MIT License](#)

Office RibbonX Editor - Version: 1.7.1 - 23 August 2020

<https://github.com/fernandreu/office-ribbonx-editor>

Written By: [Fernando Andreu](#)

Copyright © 2019-2020, Fernando Andreu

Office RibbonX Editor is free software licensed under the [MIT License](#)

Rubberduck - Version: 2.5.0.5244 - 21 December 2019

Written By: [Mathieu Guindon](#)

<https://rubberduckvba.com/>

Copyright © 2020 [Rubberduck Project Contributors](#)

Rubberduck is free software licensed under the [GPL-3.0 License](#)

Special thanks extended to:

- Martin James - *Original Inspiration and mentor*
- Kyle Tuft, Roland Chee, Matthias Roth, William Lee - *Ideas/suggestions*
- Ron de Bruin - *Indispensable VBA tutorials on [Excel Automation](#), and [Mac Excel Automation](#)*
- Paul Kelly - [Excel Macro Mastery](#) YouTube videos on how to improve VBA performance
- Stack Overflow - *solutions to so many problems*

TABLE OF CONTENTS

TABLE OF CONTENTS	4
FOREWARD	8
INTRODUCTION.....	9
GETTING STARTED	12
Required Tools	12
Install Graphviz.....	12
Edit the System PATH Environment Variable.....	17
Perform Graphviz Command Line Configuration and Test	20
Install the Relationship Visualizer Excel Spreadsheet Template	23
RELATIONSHIP VISUALIZER EXCEL WORKBOOK.....	25
Overview.....	25
Worksheets.....	25
"data" Worksheet	25
"graph" Worksheet.....	25
"styles" Worksheet	25
"style designer" Worksheet	26
"sql" Worksheet	26
"source" Worksheet.....	26
"settings" Worksheet.....	27
"HELP - shapes" Worksheet	27
"HELP - colors" Worksheet	27
"HELP - attributes" Worksheet	27
"about..." Worksheet	27
Fluent UI Ribbon Tabs	28
"Graphviz" Tab.....	28
"Style Designer" Tab.....	28
"SQL" Tab	28
"Source" Tab.....	28
"Exchange" Tab	28
VISUALIZATION TERMINOLOGY.....	29
Graph	29
Node	29
Edge	29
Undirected Graph.....	29
Directed Graph.....	29
Labels.....	29
Splines.....	30
Ports	30
Clusters / Subgraphs	31
Layout Algorithms	32

circo	32
dot	32
fdp	32
neato	32
osage	32
patchwork.....	33
sfdp	33
twopi	33
CREATING GRAPHS.....	34
Create a New Workbook from the Relationship Visualizer Template	34
Save the Workbook as a Macro-Enabled Workbook.....	35
Creating Your First Graph.....	39
Constructing Simple Graphs	42
The "Graphviz" Ribbon Tab	56
Graph to Worksheet.....	56
Graph to File	57
Algorithm.....	59
Options	61
Show/Hide Worksheets	72
CREATING RELATIONSHIP VISUALIZATIONS.....	73
Introduction	73
'Happy Days' Example.....	73
Using the 'style designer' Worksheet.....	78
The "Style Designer" Ribbon Tab	78
Using the 'styles' Worksheet.....	98
Applying Custom Styles.....	103
Modifying the Style of Individual Elements	120
Grouping Information in Clusters.....	124
Adding Native Graphviz Directives.....	127
Adding Images.....	129
Summary.....	144
CREATING VIEWS	145
Introduction	145
Views vs. Layers.....	145
Continuing Our Example with a Custom View	146
PUBLISHING GRAPHS	152
Setting Output File Options	152
Specifying the Output File Format	153
Graph to File / All Views to File	153
View the File	154
EXCHANGING DATA.....	155
The "Exchange" Ribbon Tab	156

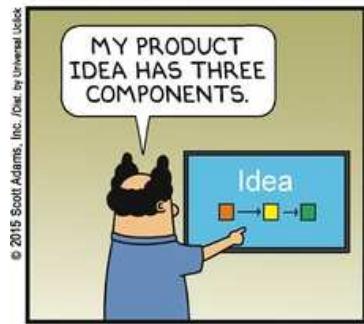
Exporting Relationship Visualizer Data to JSON format.....	157
Graphing Options	157
Workbook Metadata	158
Worksheet Layouts.....	159
'data' Worksheet.....	160
'styles' Worksheet.....	161
'sql' Worksheet.....	162
Importing JSON Data into the Relationship Visualizer.....	164
ADVANCED TOPICS, TIPS, AND TRICKS	166
HTML Labels	166
Keywords - 'Graph', 'Node', and 'Edge' Keywords	169
Clusters - Depicting a Relationship from or to a Cluster.....	171
Clusters - Aligning Nodes across Clusters	179
Nodes - Shape="Record"	182
Nodes - Shape="Record" With Ports Specified	183
Nodes - Polygon Nodes.....	184
Nodes - Skewing the Angle of Polygon Nodes.....	187
Nodes - Distorting the Length of Polygon Nodes.....	188
Edges - Consolidating Edges Using the 'strict' Option.....	189
Edges - Changing the Order of Edges.....	191
Edges - Placing a Label at the Head or Tail of an Edge.....	192
Edges - Drawing an Edge from or to the Center of a Node.....	194
Edges - Head and Tail Options.....	195
Graphs - Rotating Graphs 90 Degrees.....	199
USING SQL TO IMPORT DATA FROM OTHER EXCEL SPREADSHEETS.....	200
SQL Explained	200
ActiveX Data Objects (ADO) SQL Queries	200
SQL Ribbon Controls.....	202
SQL.....	202
Visualize	203
Writing SQL Queries to create graphs.....	203
For more information on Excel SQL	208
WORKING WITH THE DOT LANGUAGE.....	209
Viewing DOT Source Code.....	209
The "Source" Ribbon Tab	211
Source	212
Clipboard.....	212
Visualize	213
Tools	213
Web Resources	214
Working Interactively with Graphviz over the Internet.....	215
WORKING WITH LARGE DATA SETS	217
Large Data Set Example - 'The Beatles' Music Catalog.....	217

Use Views to Simplify the Graphs.....	220
Use a Data Subset to Get to Answers.....	224
CHANGING MASTER SETTINGS.....	227
Graph Options.....	227
Command Options.....	228
'data' Worksheet.....	229
'source' Worksheet	230
'sql' Worksheet.....	231
'styles' Worksheet.....	232

FOREWARD

Relationships are the way in which two or more concepts, objects, or people are connected.

Visualization is any technique for creating images, diagrams, or animations to communicate a message. Visualization through visual imagery has been an effective way to communicate both abstract and concrete ideas since the dawn of man. Examples from history include cave paintings, Egyptian hieroglyphs, Greek geometry, and Leonardo da Vinci's revolutionary methods of technical drawing for engineering and scientific purposes.



Individuals often ask to be shown the "big picture", that broad, overall perspective of a problem or situation. So much so that related common clichés or adages have found their way into our vocabulary.

- "*A picture is worth a thousand words*" refers to the notion that a complex idea can be conveyed with just a single still image. It also aptly characterizes one of the main goals of visualization, namely making it possible to absorb large amounts of data quickly.
- "*He/she can't see the forest for the trees*" refers to the inability to discern an overall pattern from a mass of detail; or to see the broader, more general situation.

If big picture views are so valued, then why are they so uncommon? The simple answer is that big picture views are hard to produce. Think of it this way, it is much easier to read a map than it is to draw a map. Often, we are drowning in volumes of data, and drawing diagrams is a difficult and time consuming task.

The **Relationship Visualizer for Excel** tool described in this manual derived from personal work to solve this conundrum. A philosophy of breaking a complex situation into descriptions of its simplest relationships guided the solution. Since people are comfortable collecting data in Microsoft Excel, it made for the ideal data entry tool. Finally, pairing Microsoft Excel with the free, open-source graphing engine Graphviz facilitated converting the data in spreadsheets into graphs and views providing alternate perspectives.

The Relationship Visualizer tool was inspired by the work of Martin F. James, and his visualization tool **Lego™ Charts for Excel** which you can learn about and obtain at <http://sourceforge.net/projects/legocharts/>. Martin's tool is a brilliant piece of work and makes an excellent companion for the Relationship Visualizer.

I hope you find this tool useful.

Jeff Long

Relationship.Visualizer@gmail.com

INTRODUCTION

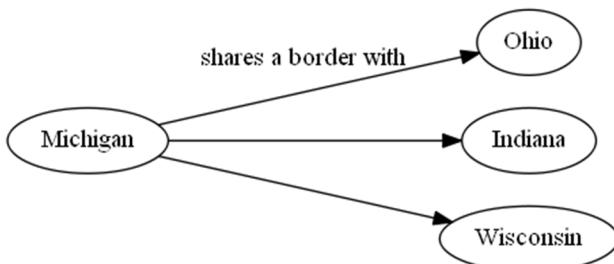
Everything around us has relationships. These relationships can be between two or more concepts, objects, people, etc. Relationships describe the state of being connected. For example, an apple is related to an apple tree; an apple tree is related to an orchard; an orchard is related to a plot of land; a plot of land is related to an owner as well as a geographical location.

It can be difficult to see the breadth of relationships because either the information about the connections is spread across many people and sources, or the volume of information makes the challenge of depicting the relationships too daunting. The Relationship Visualizer makes it easy to capture relationships using language and see the relationships in pictures using graphs.

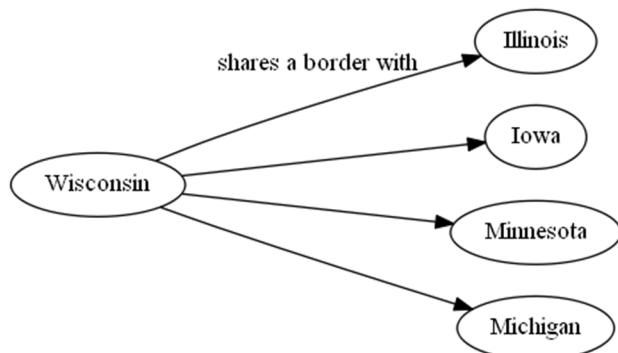
Let's look at a simple example. The states in the United States of America are related through the borders they share.

- Michigan shares a border with Ohio, Indiana, and Wisconsin.
- Wisconsin shares a border with Illinois, Iowa, Minnesota, and Michigan.
- And so on...

In businesses throughout the world people like to step up to white boards and draw these relationships as shapes and lines because it is simple to represent the concept, and as the old adages says "a picture is worth a thousand words". Using the examples above, the pictures of the border relationships described above is graphed as follows:



and



As the number of relationships grow, people tend to abandon the pictures, and revert to collecting the information in lists. Microsoft Excel is an excellent tool for this task, and people find it easy to use. To capture the state border relationships in this example, one only needs two columns in Excel, a column containing the name of the state, and another column containing the name of the state which it borders. It only takes a small amount of time to collect the information for all 50 states in a table such as:

A	B
State	Shares a Border With
1 Alaska	
2 California	Arizona
3 Florida	Alabama
4 Georgia	Alabama
5 Georgia	Florida
7 Hawaii	
8 Indiana	Illinois
9 Iowa	Illinois
10 Kansas	Colorado
11 Kentucky	Illinois
12 Kentucky	Indiana
13 Louisiana	Arkansas
14 Maryland	Delaware
15 Maryland	District of Columbia
16 Massachusetts	Connecticut
17 Michigan	Indiana
18 Minnesota	Iowa
19 Mississippi	Alabama
20 Mississippi	Arkansas
21 Mississippi	Louisiana
22 Missouri	Arkansas
23 Missouri	Illinois
24 Missouri	Iowa
25 Missouri	Kansas

Contrast entering information into Excel with illustrating the state border relationships for all 50 states in a drawing. Most people would agree that

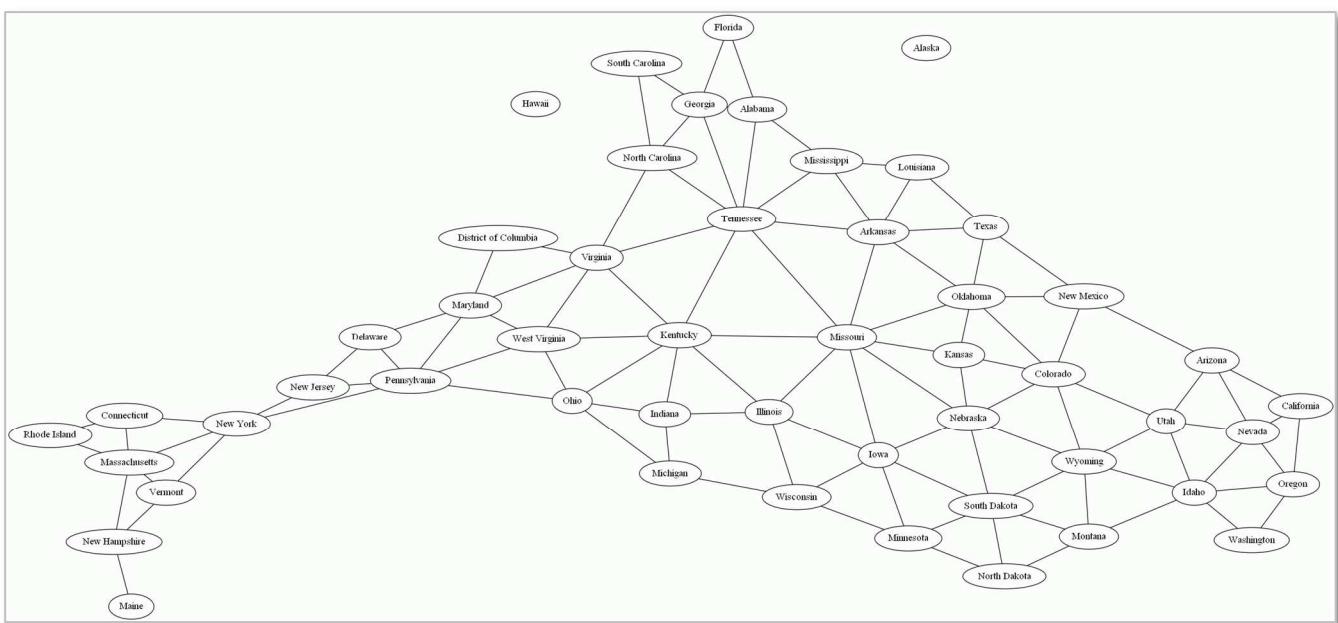
it would be a tedious, time-consuming task. First you would need to have access to a drawing tool such as Visio. Next you would need to create 50 shapes to depict each of the states. You would then need to connect all the shapes with lines, and finally you would need to arrange the shapes to optimize the drawing so that shapes are spaced evenly, and the lines do not cross over one another. Each time you move a shape it will require moving other shapes and lines until a crisp, balanced view is created. Most people are not willing to devote the time necessary to create such a drawing when the information can be collected in Excel in a fraction of the time.

While collecting the information in Excel might be expedient, some problems are easier to solve using a picture. If we were given the question "What is the least number of states you must pass through to get from Michigan to California, and

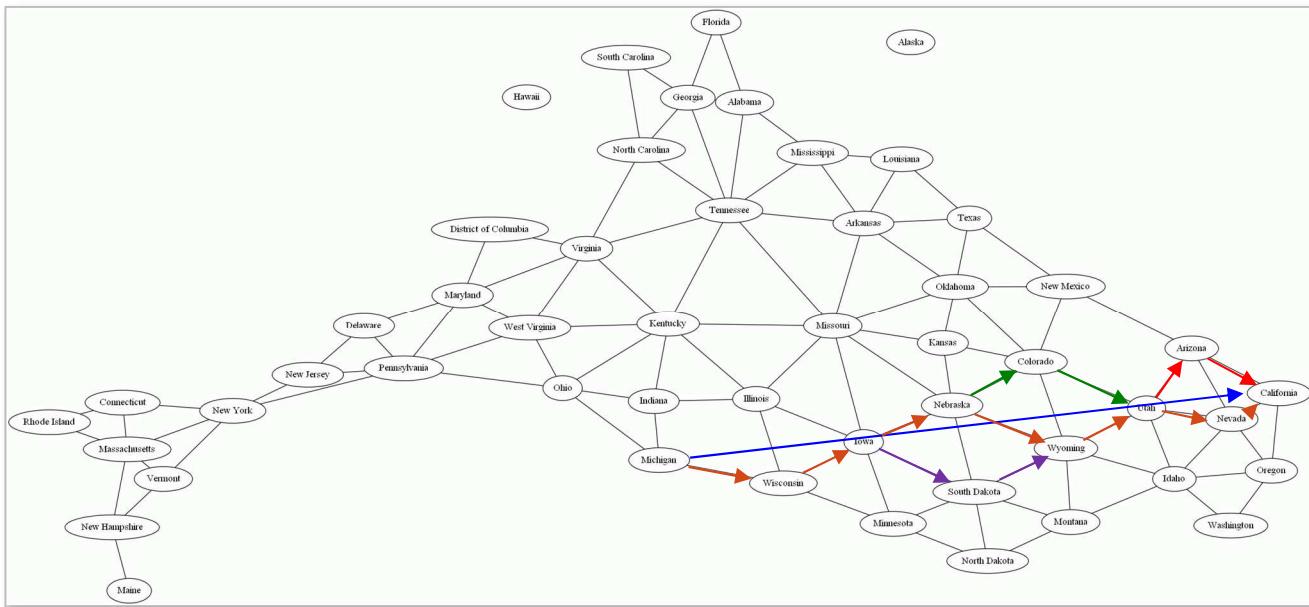
"what are the names of those states?" the answer can be found in the Excel table data, but it won't be easy. Getting the answer however will take many, many enumerations, jumping from row to row.

The easier way to solve this problem is to see the information in a graph and walk the connections with your finger. Your eyes will scan ahead to see the shortest routes, and you can count which path leads to the lowest number.

The Relationship Visualizer gives you the ability to use the ease of collecting relationship information as a list within Excel and draw relationship graphs within seconds. Copying the data from the table above into the Relationship Visualizer and asking it to generate a graph turns the data into the following visual information:



From here we can begin to determine that the route which connects Michigan and California through the least number of borders would occur approximately along a straight line from Michigan to California. This line is depicted by the blue arrow drawn on the graph below.



We can draw orange arrows on this graph to depict a route which answers the question "What is the least number of states you must pass through to get from Michigan to California?" Our eyes quickly tell us that six states are least number of states that must be crossed.

The graph shows us that there are multiple answers to the question "and what are the names of those states?" The answer we have drawn in orange arrows is "Wisconsin, Iowa, Nebraska, Wyoming, Utah, and Nevada", however scanning the graph visually we can see there are alternate portions of the route which keeps the number of states at 6, but directs you through different states. These segments are depicted by the green, red, and purple arrows.

A common thought when viewing a graph such as the one above is that it does not resemble the way that the United States are laid out geographically on a map. That is true. In this case, it the graph appears somewhat as an inverted version of the USA, with Maine's location in the Southwest, while

geographically it is in the Northeast. The reason for this depiction is that the Graphviz graphing engine does not know anything about geography. It builds the graph from the relationships and lays them out in a manner which most efficiently utilizes the diagram space so that collisions between the

shapes and lines is minimized.

By releasing the mental paradigm of where shape and line placements should occur you will reap the benefit of rapid visualizations. Give the Relationship Visualizer a bunch of simple '**A**' is related to '**B**' statements, and it will give back a big-picture graphical depiction of the data.

GETTING STARTED

Required Tools

The Relationship Visualizer is a macro-enabled Excel spreadsheet that facilitates the collection of relationship information. It works in conjunction with Graphviz, which is open-source graph visualization software¹. Graphviz's strength is the ability to generate diagrams programmatically. To fulfill this aim, Graphviz uses a simple but powerful graph description language known as DOT.

The Relationship Visualizer is an Excel spreadsheet that removes much of the burden of understanding the DOT language. It allows you to express relationships through text in Excel rows and columns. Macros in the spreadsheet write the row and column data in DOT format into a text file, and then invoke Graphviz to read the text file and interpret the commands to create the graph.

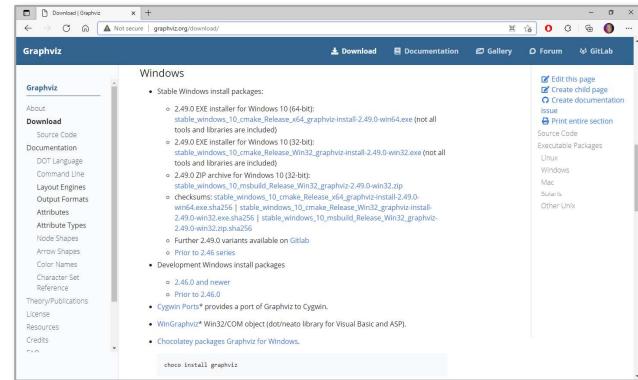
The Relationship Visualizer is not primarily an interactive tool. Given very high-level instructions, it will draw the relationship diagrams for you, considering all the low-level details such as the optimal placement of shapes and the lines that connect them. Though you have a large choice of customization options and can control the final output in many ways, it is not at all easy to force DOT to produce exactly what you want, down to the pixel. If you want to control every single pixel in your diagram, or if you are an artistic person who likes to draw free hand, then this tool is not for you.

This tool is intended for the person who wants the visualization performed with the minimum effort and without caring too much about the details of how the diagram was produced.

¹ Graphviz is an open source project developed at AT&T and released under the Eclipse Public License v1.0.

Install Graphviz

Step 1 - Download the installation media from the Windows section of the Graphviz download page at <http://graphviz.org/download/> which appears as:



You will be presented with a choice of links. Win32 for the 32-bit installer, and x64 for the 64-bit installer. Choose the link which corresponds with the Windows architecture of your machine.

Download the installer file. At the time of this writing (29-Aug-2021) the files are named:

- **2.49.0 EXE installer for Windows 10 (64-bit):** stable_windows_10_cmake_Release_x64_graphviz-install-2.49.0-win64.exe
- **2.49.0 EXE installer for Windows 10 (32-bit):** stable_windows_10_cmake_Release_Win32_graphviz-install-2.49.0-win32.exe

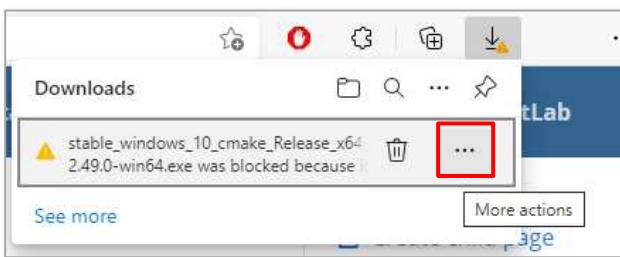
and the links above will download the version 2.49.0 installers.

If you are using the Microsoft Edge browser the download may get blocked.

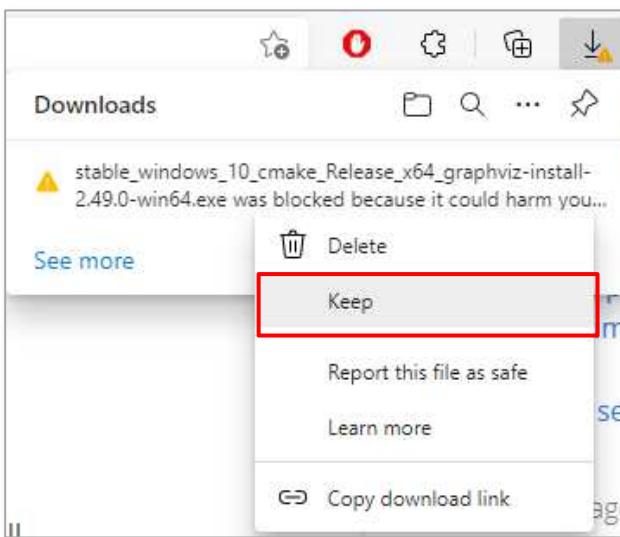
If blocked, you will see a warning icon in the download symbol. Clicking on the button provides the reason the download was blocked.



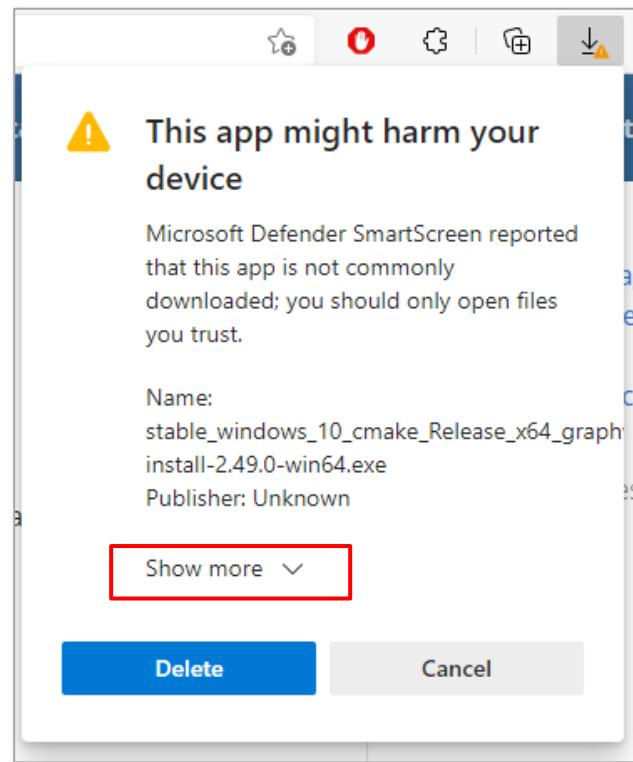
Hover your mouse over the message, and a button with three dots will appear, along with a tooltip which says, "More actions".



Click on the [...] button, and a popup menu appears. Select "Keep" from the dropdown list.

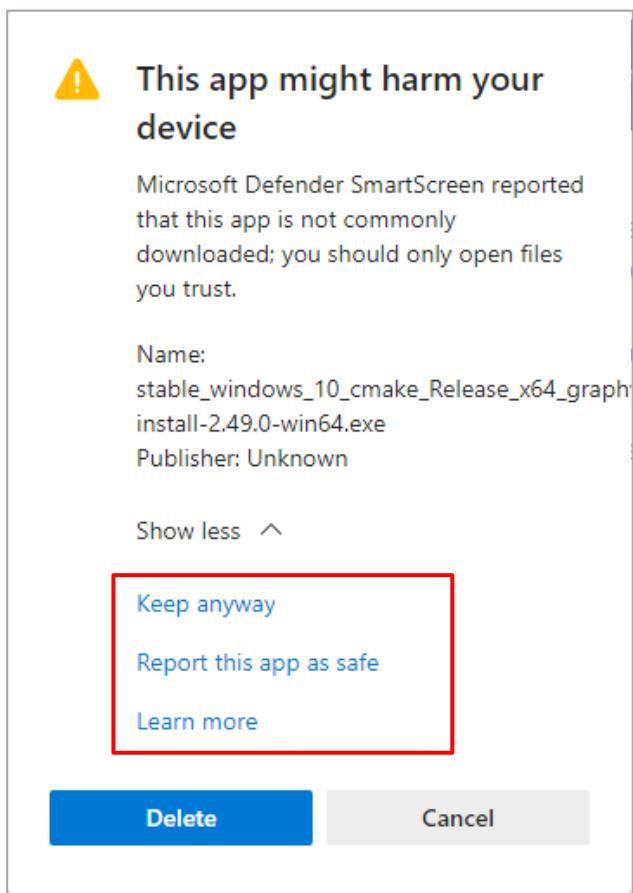


Microsoft Edge will again try to dissuade you from downloading the file with a warning "This app might harm your device". It will appear that your only choices are "Delete" or "Cancel". To keep the file you must click the "Show more" dropdown.

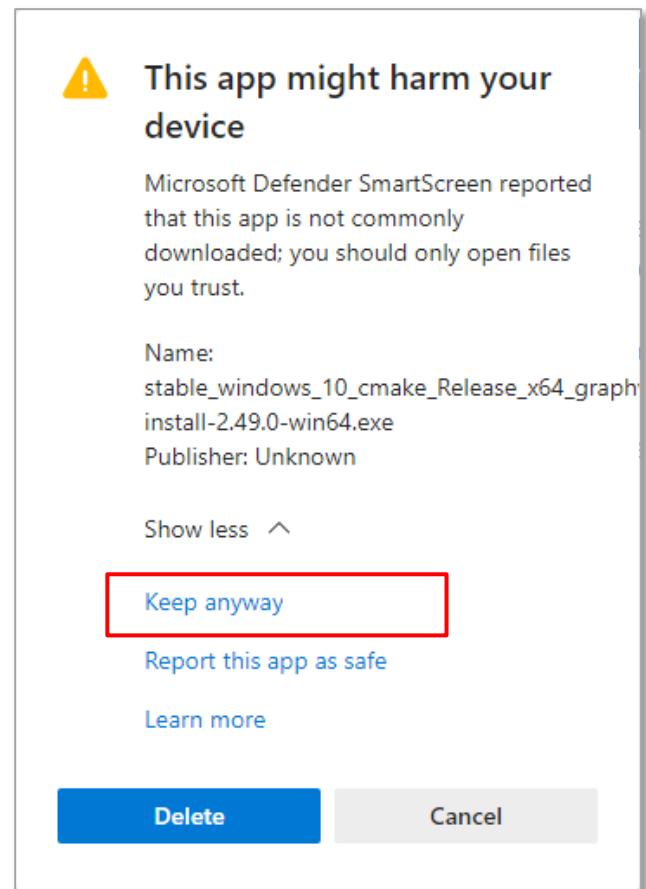


Three additional choices will now appear:

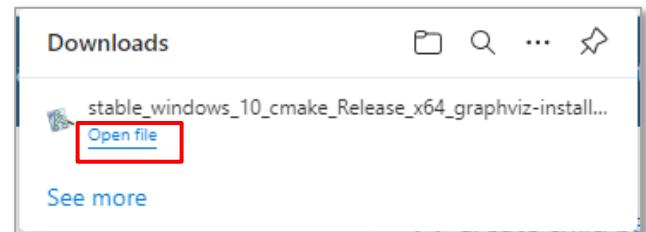
- Keep anyway
- Report this app as safe
- Learn more



Click on "Keep anyway"



Now the installer file shows up as a download.
Click on "Open file" to run the installer.



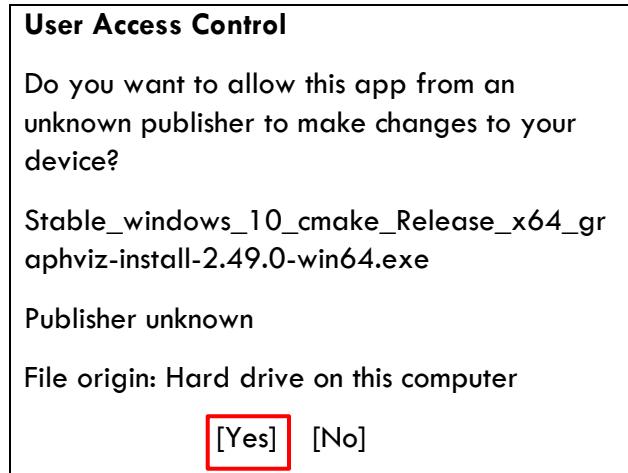
Step 2 - Launch the installer file. Once again you will receive a security warning since this file was downloaded over the internet. It will appear that your only choice is "Don't run". To run the installer, click on the "More info" link.



Information showing the name of the installer file is shown, and a "Run anyway" button appears. Click this button

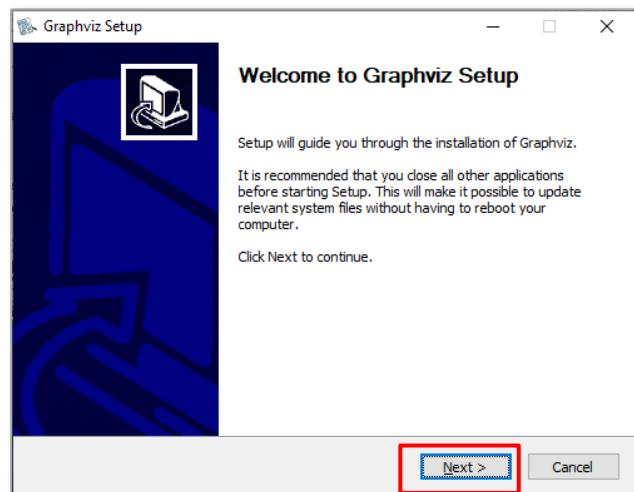


A User Access Control warning will now take over the screen and ask



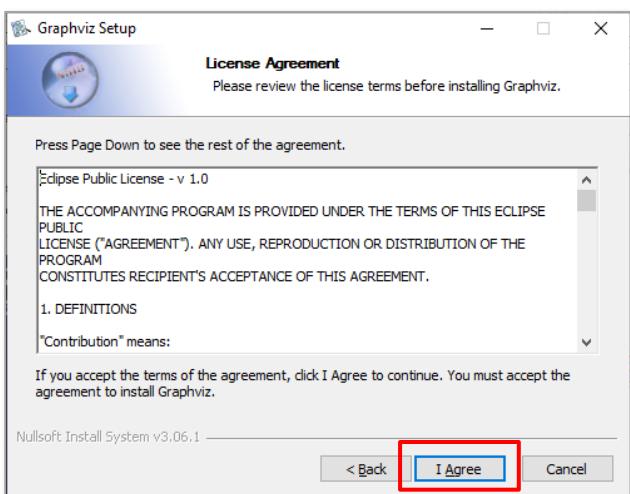
Press the Yes button, and the Graphviz installer will begin to run.

Step 3 - Select "Next >" on the "**Welcome to Graphviz Setup**" splash page.

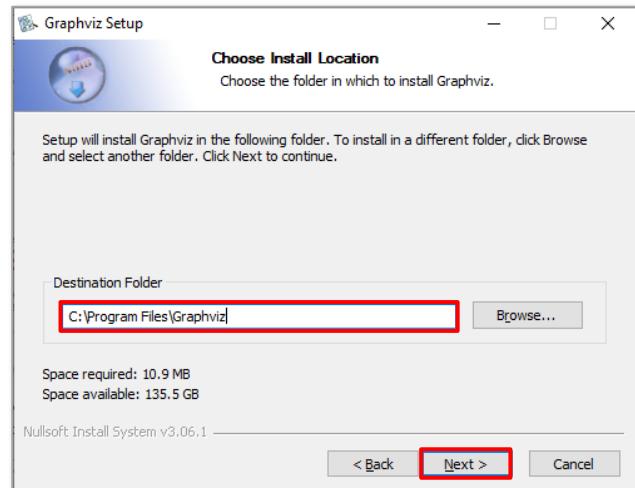


Excel to Graphviz for Win OS

Step 4 - Accept the License Agreement by pressing the "I Agree" button.

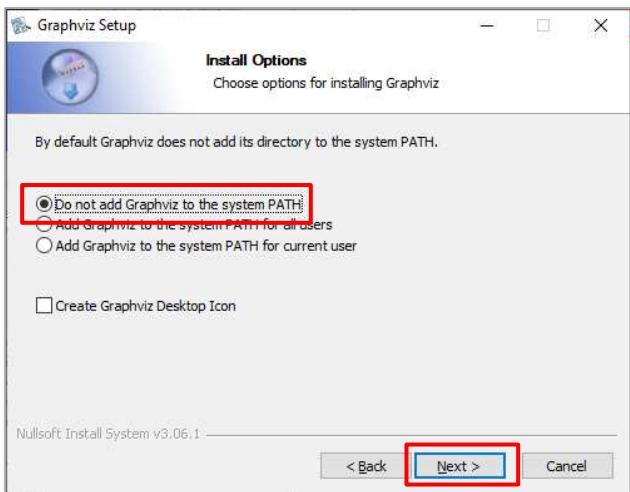


Step 6 - Specify the folder where Graphviz should be installed. Select the "Everyone" radio button, and then select the "Next >" button.

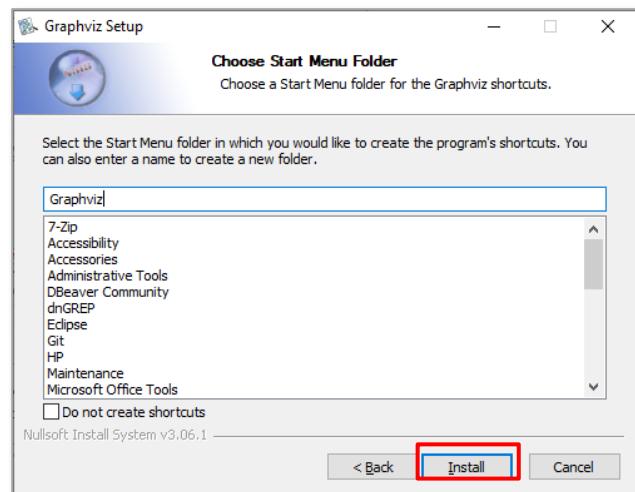


Step 5 - The Graphviz bin directory needs to be on your path. You can allow the installer to perform this action (easy way), or you can perform it manually (hard way). We will perform this task manually after the install to ensure you know how to do it should the installer encounter any problems.

Select the radio button "Do not add Graphviz to the system PATH", then click the "Next >" button.

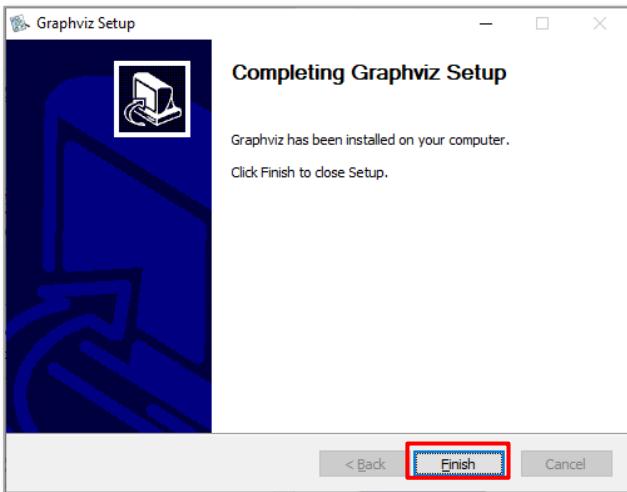


Step 7 - Choose the Start Menu Folder and press the "Install" button.

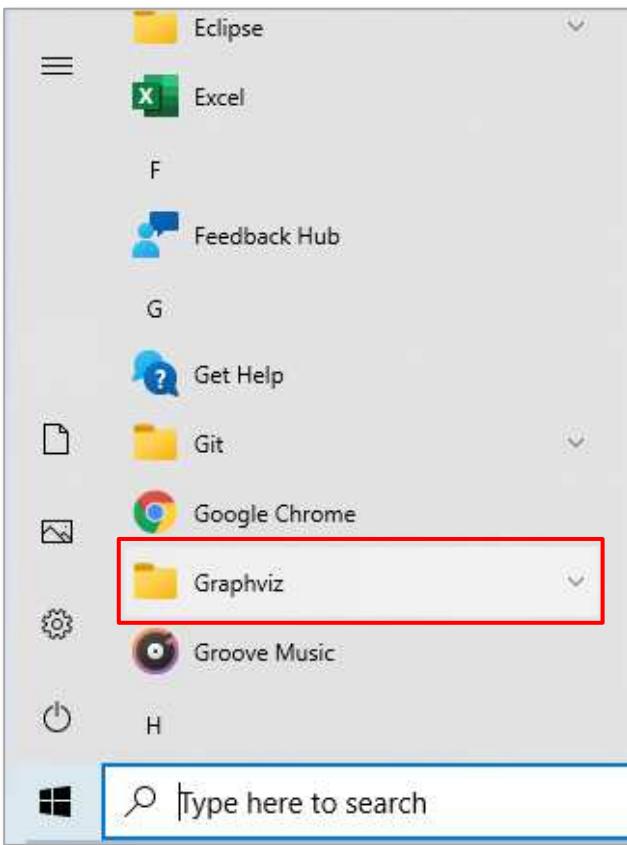


Step 8 - The installer will copy the files and make Windows configuration changes. A status bar will indicate how the installation is progressing.

Step 9 - Once the "Completing Graphviz Setup" screen appears press the "Finish" button. The software is installed.



Step 10 - Confirm Graphviz is installed. If you select the Microsoft Windows start button you should see "**Graphviz**" as a program folder.

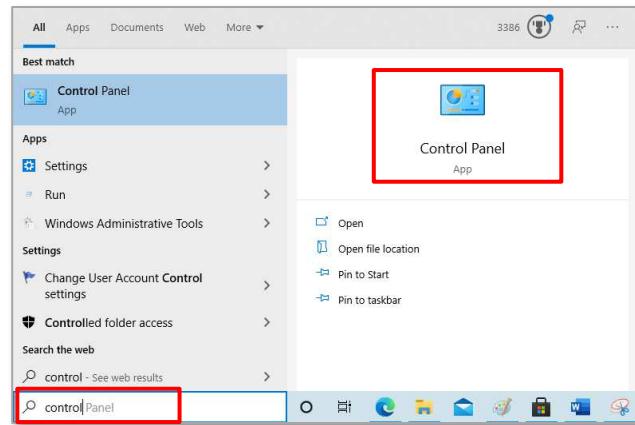


Edit the System PATH Environment Variable

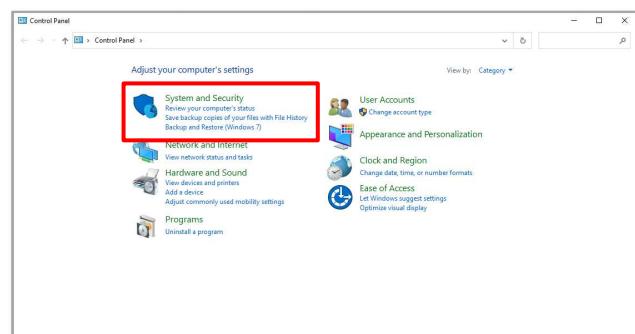
The Relationship Visualizer utilizes the command line programs provided by Graphviz. The Graphviz installer can modify your PATH environment variable, however if your path is too long the installer may fail. It is your responsibility to perform this task manually when following these instructions.

The detailed steps illustrated using a Microsoft Windows 10 operating system are as follows:

Step 1 - Click the Start button and begin to type "Control Panel". Select the Control Panel icon when it appears.

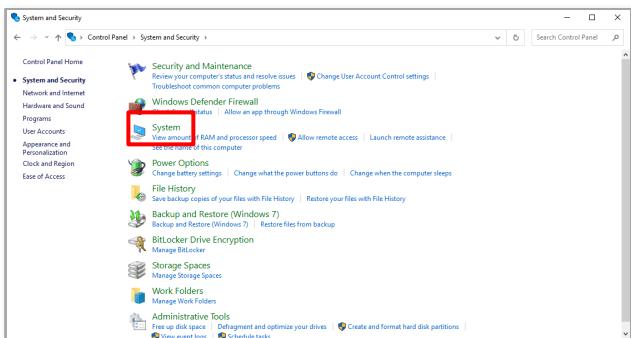


Step 2 - Click on "System and Security"

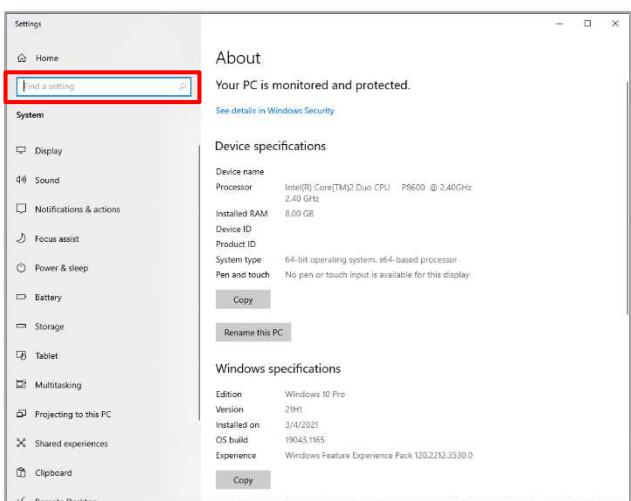


Excel to Graphviz for Win OS

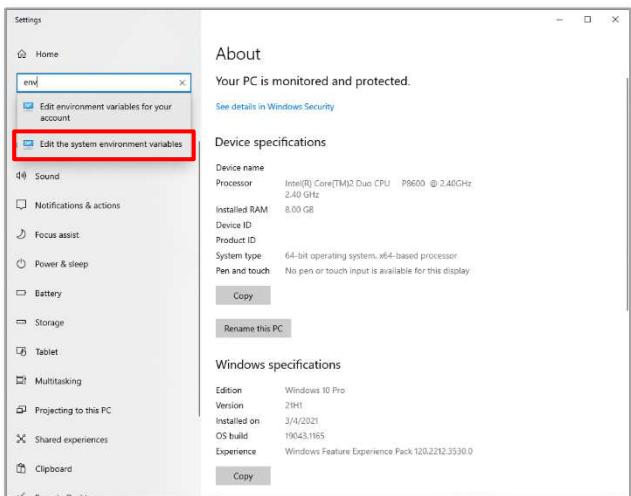
Step 3 - Click on "System"



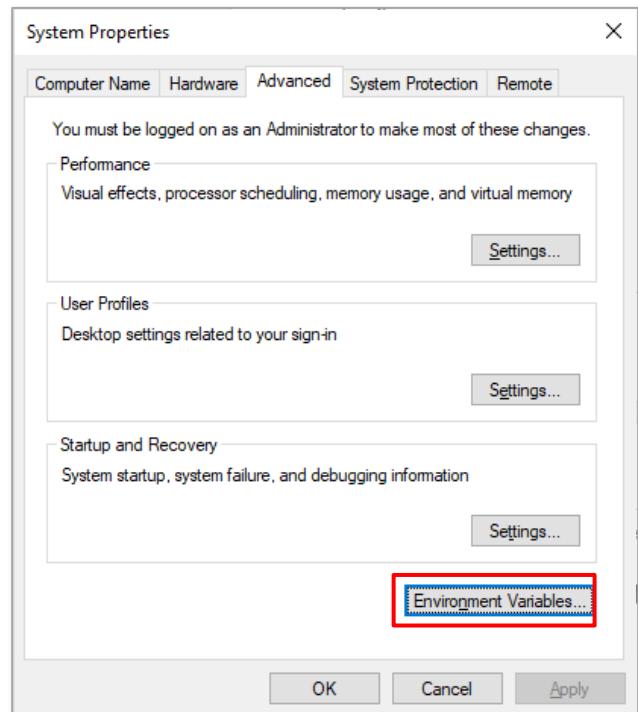
Step 4 – Under “Home”, begin to type “Environment Variables” in the search box “find a setting”.



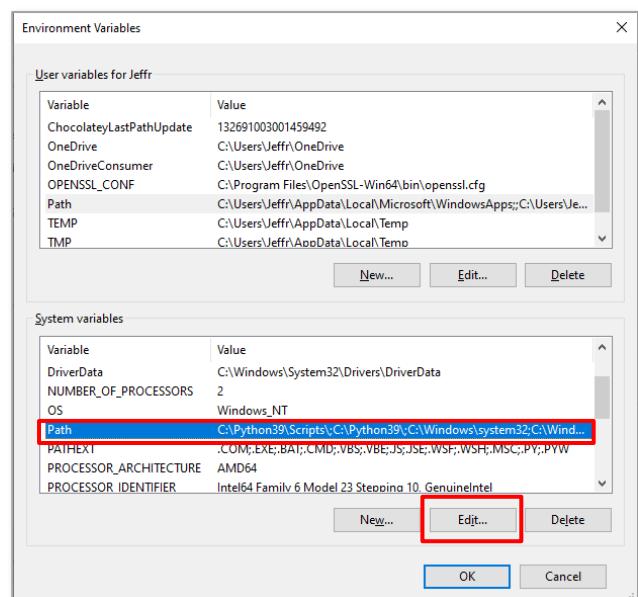
Choices beginning with “Env” will begin to appear.
Select “Edit the system environment variables”.



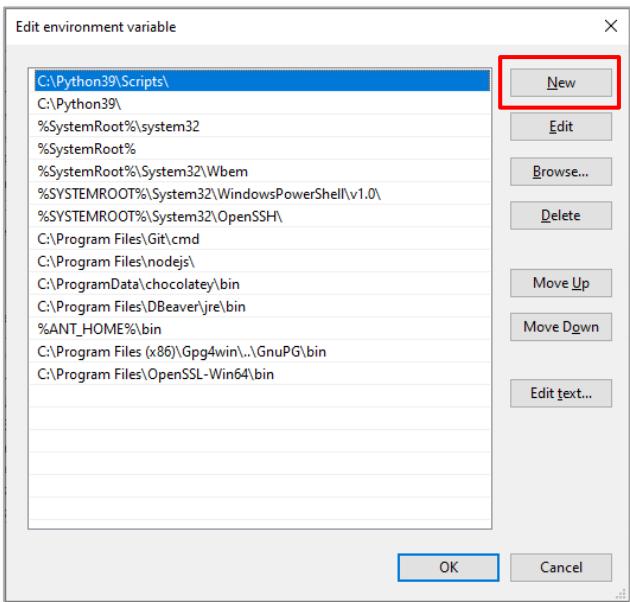
Step 5 - Select the "Environment Variables..." button.



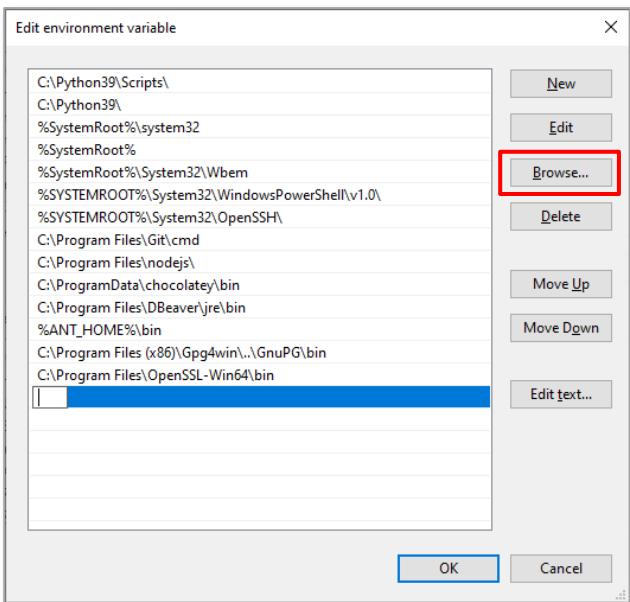
Step 6 – “Path” appears in the “User variables” as well as the “System variables”. Highlight the “Path” line in the “Systems variables” list (the bottom list), then select the “Edit...” button.



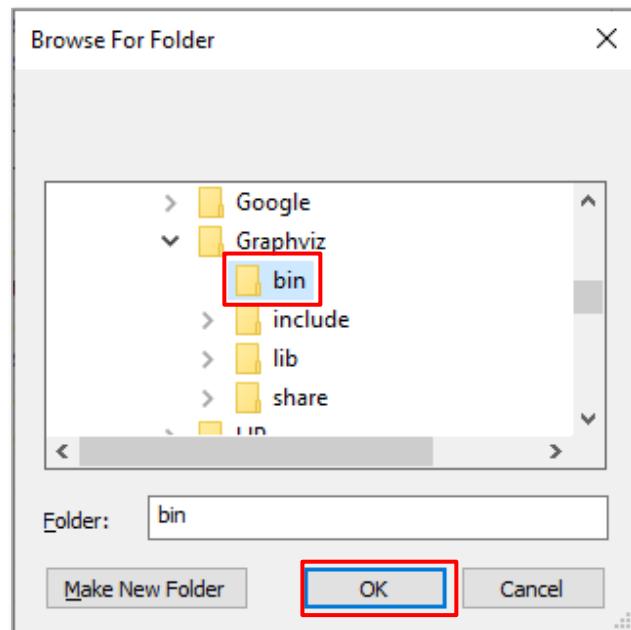
Step 7 - The "Edit environment variable" dialog appears. Press the "New" button.



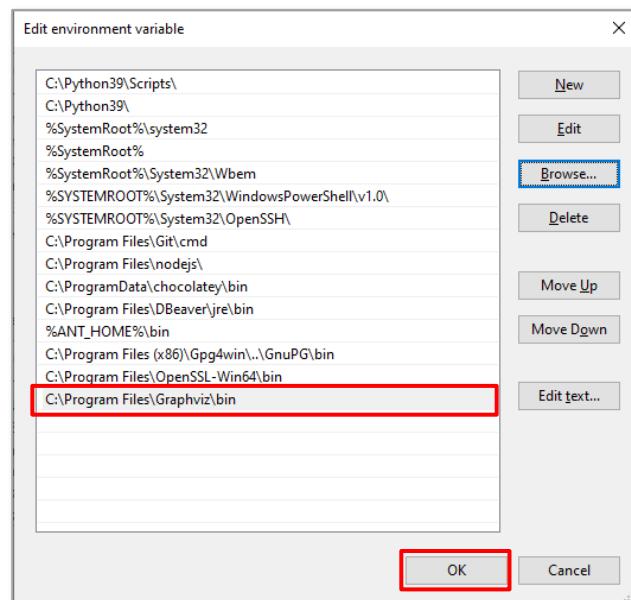
Step 8 - A new line is added at the bottom of the list. Press the "Browse..." button.



Step 9 - The "Browse for Folder" dialog appears. Navigate to the Graphviz bin directory, then press the OK button.

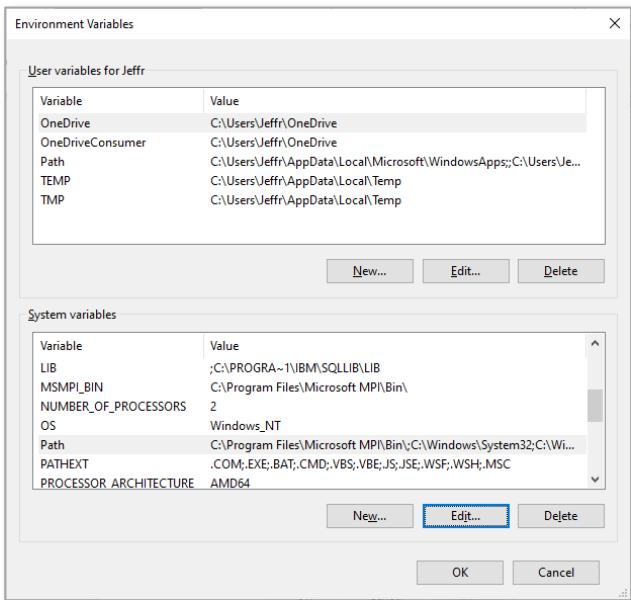


If your list is long you may want to use the "Move Up" button to move the directory up in the PATH. Press the OK button.



Excel to Graphviz for Win OS

Step 11 - You are returned to the "Environment Variables" dialog. Press the OK button to commit the environment variable changes and close the dialog.



Step 12 - (Optional) Restart Microsoft Windows.

Technically this should not be necessary, but if you have already been running Excel there is the possibility that it may be holding an old copy of your environment variables. Restarting Windows will ensure that Excel will reference a current PATH environment variable.

Perform Graphviz Command Line Configuration and Test

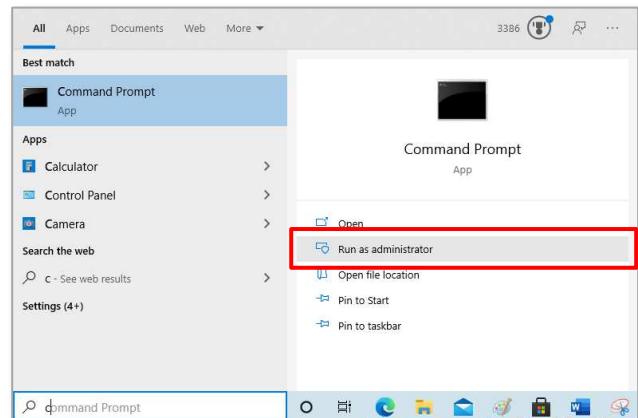
At this point, you have completed the installation steps to install the Graphviz software.

The Relationship Visualizer spreadsheet uses the command line programs to generate the graph visualizations. You must manually execute a command line command to configure the Graphviz plugins before you can use Graphviz properly.

NO NOT SKIP THIS STEP!

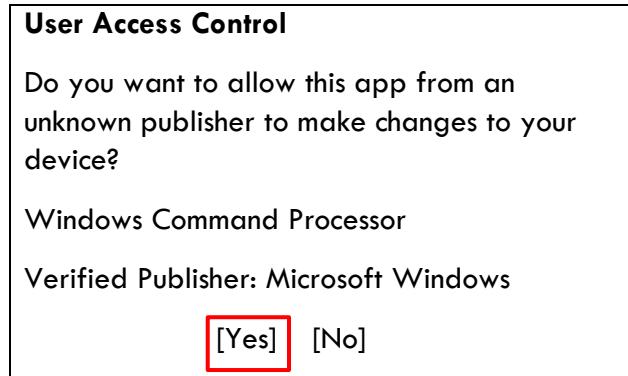
Testing the command line programs prior to using the spreadsheet can help ensure that everything is in place correctly so that the spreadsheet can perform properly.

Step 1 - Open a Command Prompt window using the "Run as Administrator" option. Click on the Windows Start Menu icon and begin to type Command Prompt. When the Command Prompt App appears choose the "Run as administrator" option.

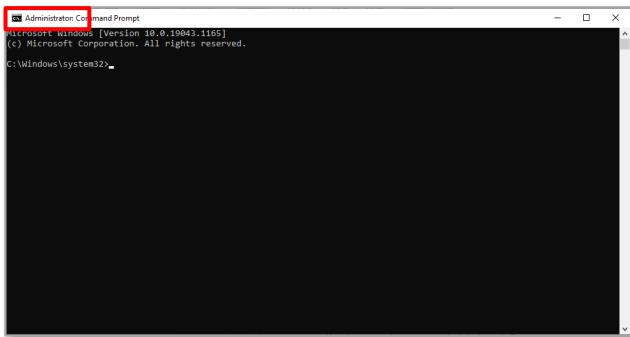


Step 2 - You will get asked for permission to run a command prompt as Administrator.

Press the "Yes" button.



Step 3 - A command prompt window appears. Confirm that the word "Administrator" appears in the Window title.



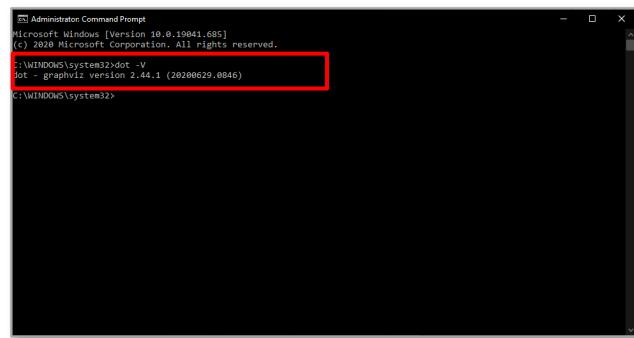
Step 4 - Test that Graphviz is configured properly on the PATH by entering the command:

dot -V

noting that the `-V` switch (for version) should be in uppercase, not lowercase. The dot program should respond with the message:

```
dot - graphviz version
2.49.0 (20210828.1703)
```

in similar fashion to the screen print below:



If you receive the message:

'dot' is not recognized as an internal or external command, operable program or batch file.

It means one of the following things:

- You have specified the path to the Graphviz executables incorrectly and you should repeat the steps above. Things to check are:
 - Did you remember to add the `\bin` subdirectory to the Graphviz directory path?
 - Is the directory placed at the end of the PATH such that the length of the PATH exceeds the Windows length limit? If so, move the Graphviz bin directory closer to the beginning of the list.
- You opened the Command Prompt window prior to updating the PATH statement. This command window is still recognizing the old path. Close the Command Prompt window, open a new one, and repeat the `dot -V` command.

Step 5 - Configure the plugins by entering the command

```
dot -c
```

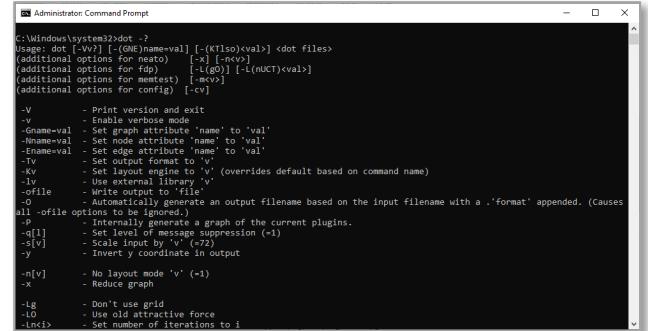
This is the important step which must not be skipped.

No messages are written when the command executes; the screen will look as follows:



```
C:\Windows\system32>dot -c
C:\Windows\system32>
```

The screen will appear as follows:



```
C:\Windows\system32>dot -?
Usage: dot [-Vv?] [-GNAME=name=val] [-(KTlso)<val>] <dot files>
(additional options for graph) [-TTYPE] [-NNAME] [-I(NCT)<val>]
(additional options for edges) [-EFORMAT] [-L(LABEL)] [-I(INKT)<val>]
(additional options for nodes) [-NFORMAT] [-M(METHOD)<val>]
(additional options for config) [-cv]

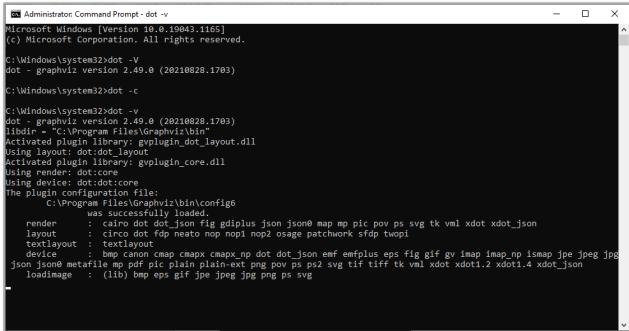
-v          - Print version and exit
-V          - Enable verbose mode
-GNAME=val - Set graph attribute 'name' to 'val'
-HNAME=val - Set graph attribute 'height' to 'val'
-ENAME=val - Set edge attribute 'name' to 'val'
-TTYPE      - Set output format to 'T'
-X          - Set layout mode to 'X' (overrides default based on command name)
-Y          - Use external lines for 'Y'
-OFILE      - Write output to 'file'
-O          - Automatically generate an output filename based on the input filename with a .format appended. (Causes all .ofile options to be ignored)
-p          - Internally generate a graph of the current plugins.
-[I]        - Set level of message suppression (-1)
-[V]        - Scale input by 'V' (-72)
-y          - Invert y coordinate in output
-[V]        - No layout mode 'V' (-1)
-x          - Reduce graph
-LG         - Don't use grid
-LO         - Use old attractive force
-LNCi       - Set number of iterations to i
```

Congratulations! Graphviz is installed properly.

Step 6 - To see the list of configured plugins type the command

```
dot -v
```

where the -v is lowercase. The screen will appear as follows:



```
C:\Windows\system32>dot -v
Microsoft Windows [Version 10.0.19043.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>dot -v
dot - graphviz version 2.49.0 (20210828.1703)
C:\Windows\system32>dot -c

C:\Windows\system32>dot -v
dot - graphviz version 2.49.0 (20210828.1703)
libdir : "C:\Program Files\Graphviz\bin"
Activated plugin library: gvplugin_dot_layout.dll
Using layout: dot_layout
Activating plugin library: gvplugin_core.dll
Using render: dotcore
Using device: dotdotcore
The plugin configuration file:
  C:\Program Files\Graphviz\bin\config6
  was successfully loaded.
render   : cairo dot dot_json fig gdplus json jsonn map mp pic pov ps svg tk vml xdot xdot_json
layout   : cisco dot dpd neato nop nopl nopt usage patchwork sfdp twopi
textlayout : textlayout
device   : bmp canon cmap cmapx_np dot dot_json emf emfplus eps fig gif gv imap imap_np ismap jpeg jpg
json jsonn metatile mp pdf pic plain plain-ext png pov ps2 svg tikz tiff tk vml xdot xdoti_2 xdoti_4 xdot_json
loadimage : (lib) bmp eps gif jpe jpg png ps svg
```

At this point Graphviz is waiting for more input. Hitting the **Ctrl** key + **C** key will break you from the **dot** program.

Step 7 - To see the list of command line options you can enter the command

```
dot -?
```

Install the Relationship Visualizer Excel Spreadsheet Template

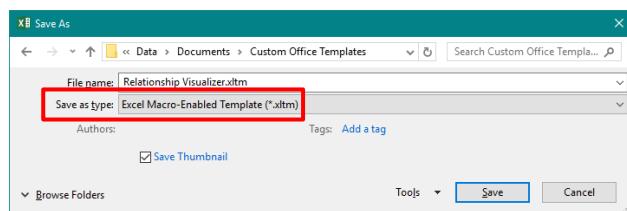
Step 1 - In the root directory of the Relationship Visualizer distribution there is a macro-enabled Excel spreadsheet "Relationship Visualizer.xlsx". Double-click the mouse on the file to launch Excel. You will probably get a security warning that the spreadsheet contains macros. You need to enable macro support to use the Relationship Visualizer spreadsheets.



Step 2 - Perform a "File > Save As" operation. When it asks you where to store the file, navigate to the directory where you currently have the template file (This PC > Documents > Custom Office Templates).

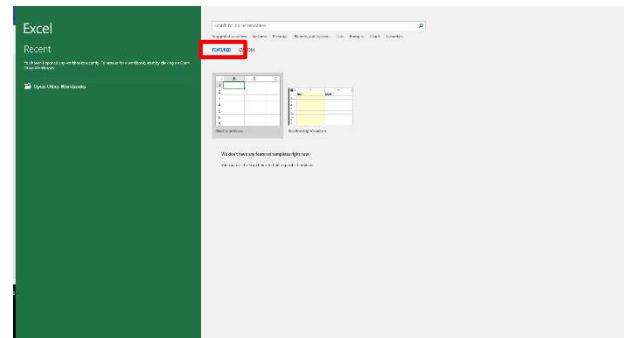
Next, you will see the file name is "Relationship Visualizer1". Change it back to "Relationship Visualizer".

Where it says "Save as type:" select "**Excel Macro-Enabled Template**" from the dropdown list. You will notice that the save location will change to your personal "Custom Office Templates" directory. Select OK, and Excel will place a copy of the template file into this directory.

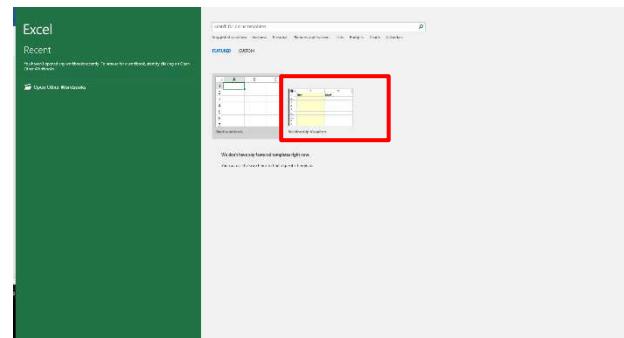


Step 3 - Close Excel.

Step 4 - Launch Excel. Excel will offer a selection of built-in spreadsheet templates you can use. Look under the title "FEATURED" for the Relationship Visualizer template.

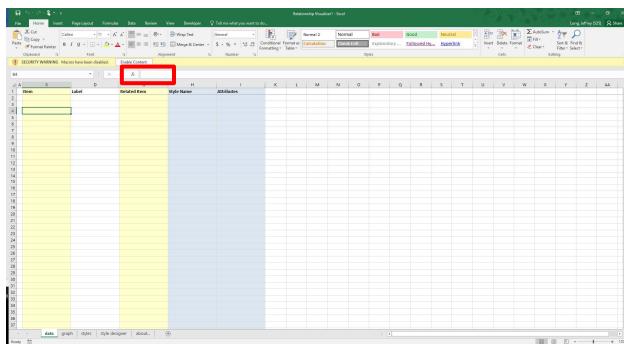


Step 5 - The Relationship Visualizer template will be listed, along with a thumbnail image. Click on.

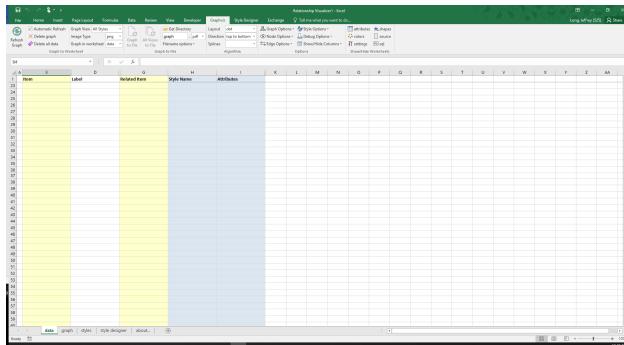


Excel to Graphviz for Win OS

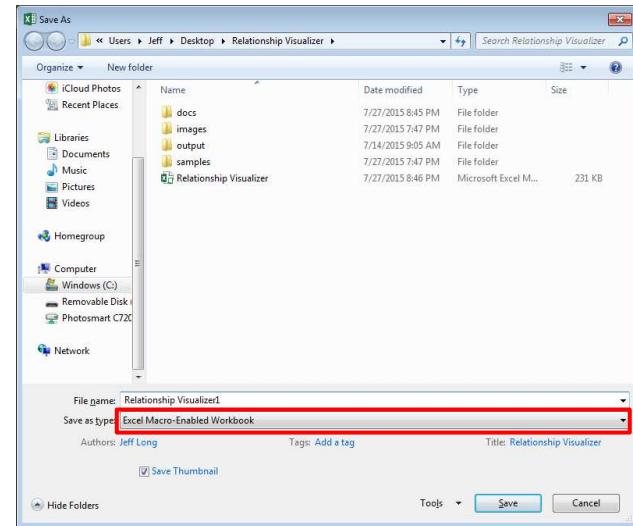
Step 6 - Excel will create a file named "Relationship Visualizer1" with a warning that the spreadsheet contains macros. Click on the "Enable Content" button to permit the macros to run.



The SECURITY WARNING bar should disappear.



Step 7 - Perform a "FILE -> Save As" action to save the file as an "**Excel Macro-Enabled Workbook**". Notice that last time we saved it as an "Excel Macro-Enabled Template" but going forward we will create new spreadsheets, populate them with data, and save them as macro-enabled workbooks. You may save the workbook in the location of your choice, and under the name of your choice. The important thing is saving the file with the .xlsm file extension.



RELATIONSHIP VISUALIZER EXCEL WORKBOOK

Overview

The Relationship Visualizer tool is a macro-enabled Excel workbook. It allows you to collect data in Excel's familiar table format and use Excel's wide set of functions.

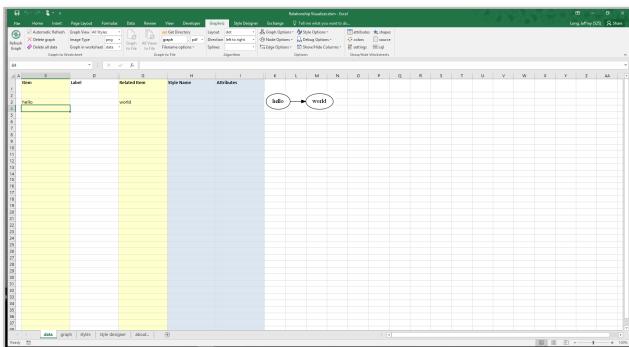
The Relationship Visualizer uses multiple worksheets to accomplish its mission. Some worksheets provide the basic functions, while others assist in creating advanced visualizations, or presentation customizations. The sections that follow provide a basic overview of the worksheets to help you navigate the workbook. Later chapters describe how to use each worksheet.

Worksheets

"data" Worksheet

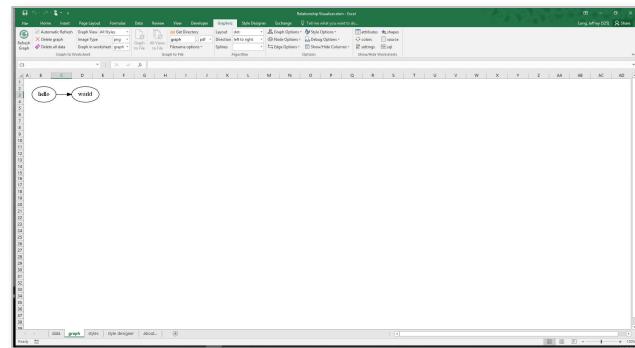
The "data" worksheet is the core of the Relationship Visualizer. This is the place where you will list the nodes and edge relationships to build your visualizations.

This worksheet is described in more detail in the section **Creating Your First Graph** beginning on page 39.



"graph" Worksheet

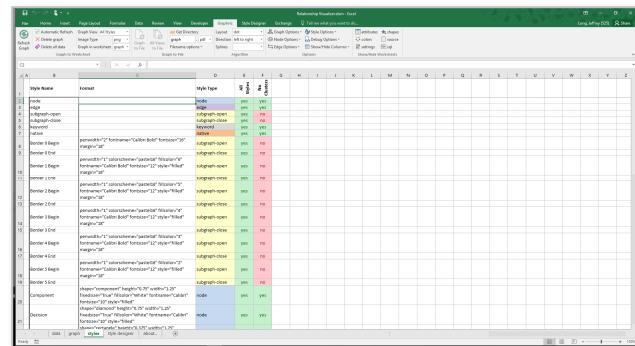
The "graph" worksheet is where image representations of data in the "data" worksheet is displayed whenever the "Refresh Graph" button is pressed within the workbook.



"styles" Worksheet

The "styles" worksheet is where you can create style definitions for nodes and edges. It is in a manner like an HTML Cascading Style Sheet where you can define a style name, and how the style should appear (shape, color, font, etc.). A defined style can then be associated with many nodes or edges in the "data" worksheet.

This worksheet is described in more detail in the section **Using the 'styles' Worksheet** beginning on page 98.

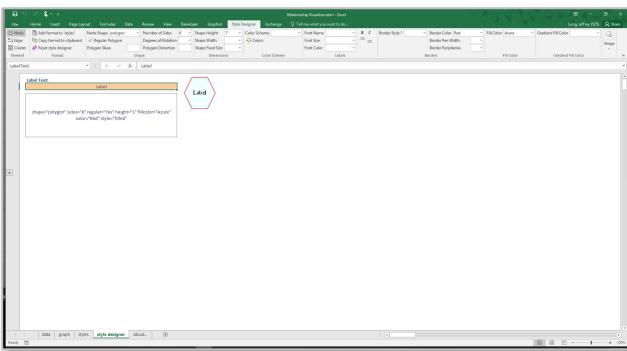


"style designer" Worksheet

The "style designer" makes it easy for you to create style definitions for nodes and edges. It removes the burden of knowing the Graphviz attributes required to achieve a visual effect. The style designer provides the capability to change settings using dropdown lists, and see a preview example of how Graphviz renders the node, edge, or cluster from the attributes.

The style designer lets you adjust the attributes until you are happy with the look, then allows you to save the style definition in the "styles" worksheet.

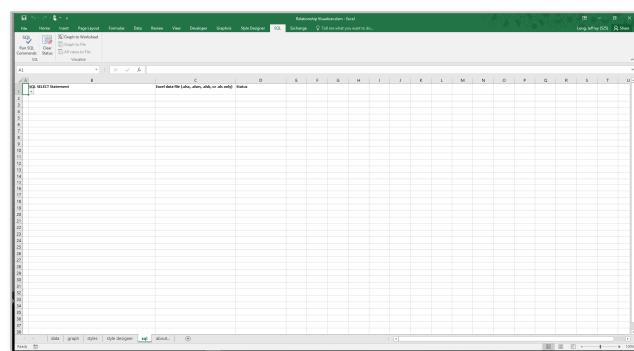
This worksheet is described in the section **Using the 'style designer' Worksheet** beginning on page 78.



"sql" Worksheet

The "sql" worksheet provides capabilities to run Excel ADODB SQL statements to bring data from external Excel spreadsheets into the "data" worksheet for graphing.

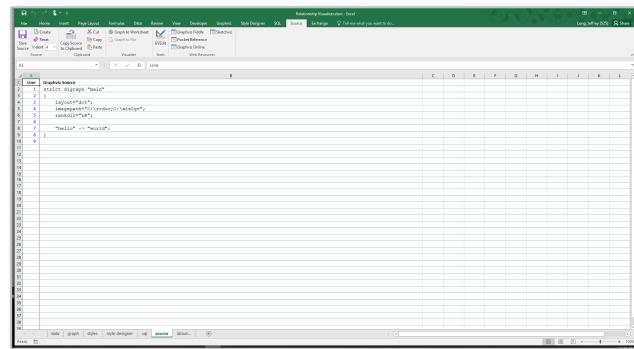
This worksheet is described in the section **Using SQL to Import Data from other Excel Spreadsheets** beginning on page 200.



"source" Worksheet

The "source" worksheet is where you can see the DOT language source code created from the data in the "data" worksheet when a graphing button was pressed, which in-turn resulted in the graph in being created. The worksheet also contains links to Graphviz language editing/rendering tools that for experimenting with the DOT language.

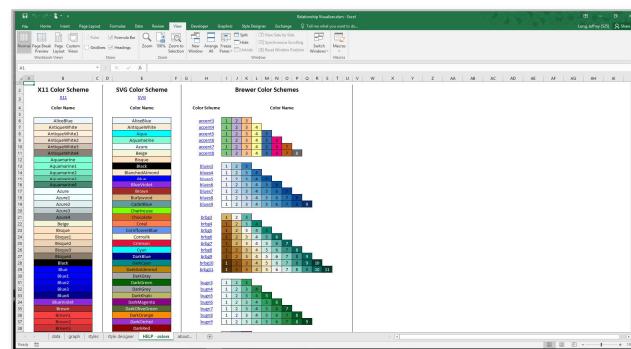
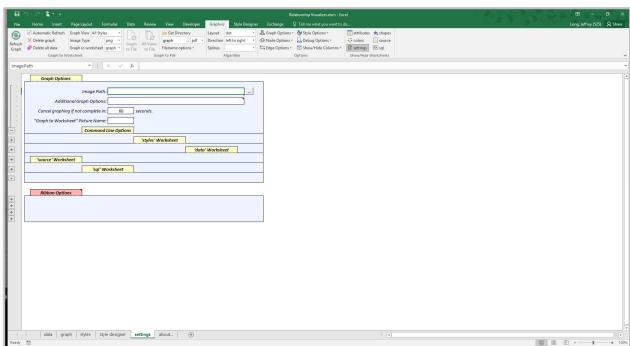
This worksheet is described in the section **Viewing DOT Source Code** beginning on page 209.



"settings" Worksheet

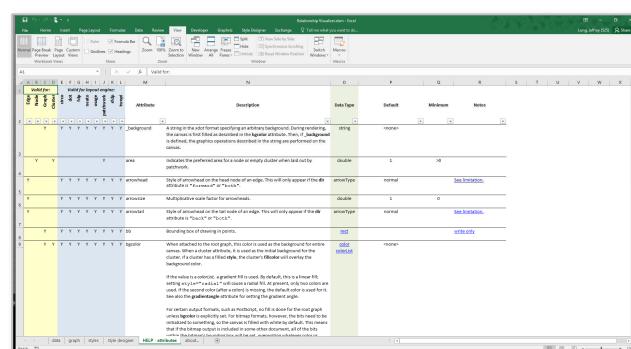
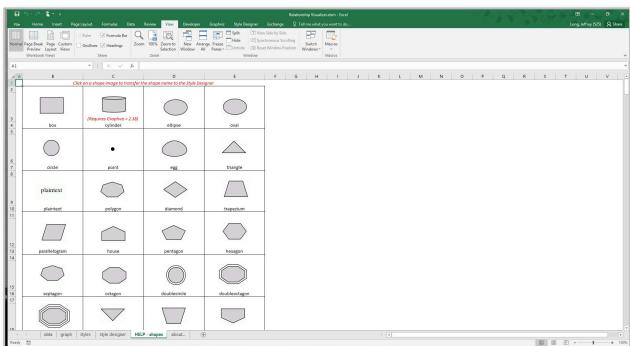
The "settings" worksheet provides capabilities to customize how Graphviz runs.

This worksheet is described in the section **Changing Master Settings** beginning on page 227 .



"HELP - shapes" Worksheet

The "HELP - shapes" worksheet is intended to provide a glossary of the node shapes which Graphviz supports along with the shape names. This reference can be of great assistance when using the "style designer" capabilities.

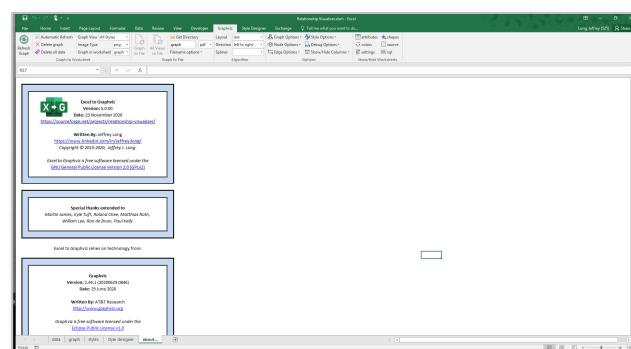


"HELP - colors" Worksheet

The "HELP - colors" worksheet is intended to provide a glossary of the color schemes and color names which Graphviz uses and what the color looks like. This reference can be of great assistance when using the "style designer" capabilities.

"about..." Worksheet

The "about..." worksheet provides the Relationship Visualizer version number, contact information for reaching the program's author, and licenses of the Relationship Visualizer and the open-source components it incorporates.



Fluent UI Ribbon Tabs

The Office Fluent ribbon replaced Microsoft's previous system of layered menus, toolbars, and task panes from previous versions of Office. The ribbon has a simpler system of interfaces that is optimized for efficiency and discoverability. The ribbon has improved context menus, screen tips, a mini toolbar, and keyboard shortcuts that improve user efficiency and productivity.

The Relationship Visualizer spreadsheet has extended Excel's ribbon interface with 5 additional tabs for performing Relationship Visualizer related actions.

"Graphviz" Tab

The 'Graphviz' tab of the Excel Fluent UI ribbon provides action buttons and run-time graph option choices.

See the section **The "Graphviz" Ribbon Tab** beginning on page 56 for more details.



"Style Designer" Tab

The 'Style Designer' tab of the Excel Fluent UI ribbon provides action buttons and run-time option choices pertaining to the 'style designer' worksheet.

See the section **The "Style Designer" Ribbon** beginning on page 78 for more details.



"SQL" Tab

The 'SQL' tab of the Excel Fluent UI ribbon provides action buttons and run-time option choices pertaining to the 'sql' worksheet.

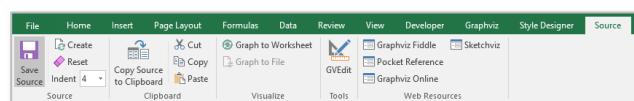
See the section **SQL Ribbon Controls** beginning on page 202 for more details.



"Source" Tab

The 'Source' tab of the Excel Fluent UI ribbon provides action buttons and run-time option choices pertaining to the 'source' worksheet.

See the section **The "Source" Ribbon Tab** beginning on page 211 for more details.



"Exchange" Tab

The 'Exchange' tab of the Excel Fluent UI ribbon provides action buttons and run-time option choices pertaining to the Export and Import of Relationship Visualizer data utilizing text files in the JSON format.

See the section **The "Exchange" Ribbon** beginning on page 156 for more details.



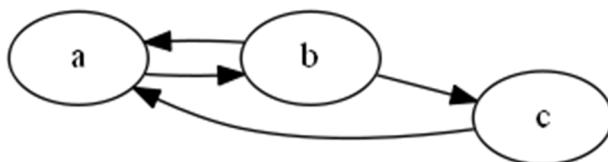
VISUALIZATION TERMINOLOGY

In mathematics and computer science, *graph theory* is the study of graphs, which are mathematical structures used to model pairwise relations between objects.

The terms described in this chapter are used throughout the rest of this manual to explain how to construct visualizations. These terms have their roots in graph theory and/or the Graphviz tool.

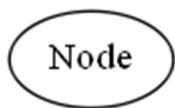
Graph

The following picture illustrates a "graph".



Node

A "graph" in this context is comprised of "nodes".



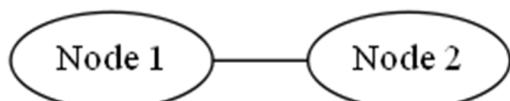
Edge

"Edges" are lines that connect nodes.



Undirected Graph

A graph may be "undirected", meaning that there is no distinction between the two nodes associated with each edge.



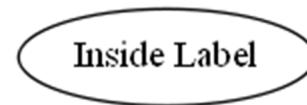
Directed Graph

A graph may be "directed" meaning that there is an explicit direction from one node to another.



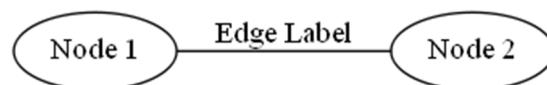
Labels

Nodes can have "labels". Labels can be placed inside the node, and outside the node.

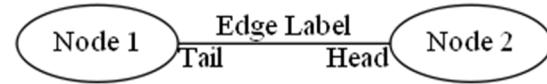


Outside Label

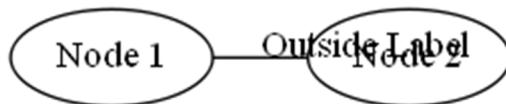
Edges can also have labels. Edge labels can be placed on the edge,



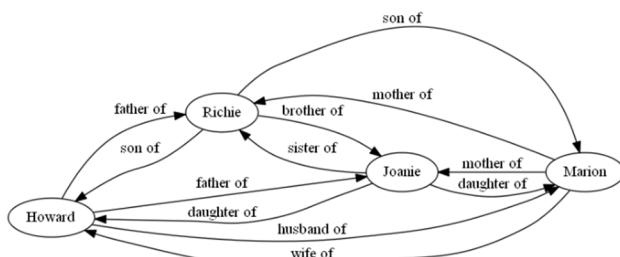
at the tail and/or head of the edge,



Or outside the edge (however in my experience they tend to not always render well)



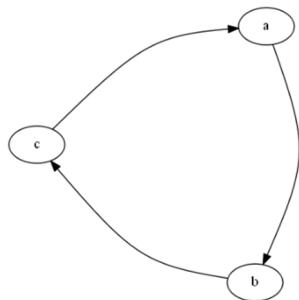
Edge labels are helpful in stating what the relationship between the nodes is. For example, a set of family relationships might look as follows:



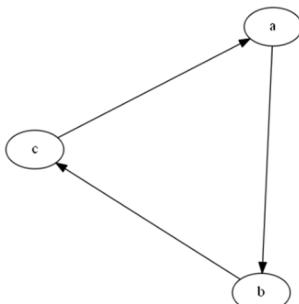
Splines

The way in which edges are routed and drawn are called "splines". Several spline types are available in the Relationship Visualizer. The spline type and a depiction of each follows:

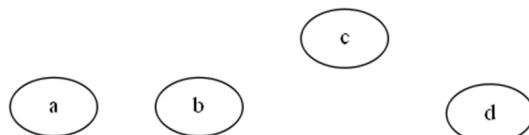
- **Curved** - Edges are drawn as curved arcs between nodes



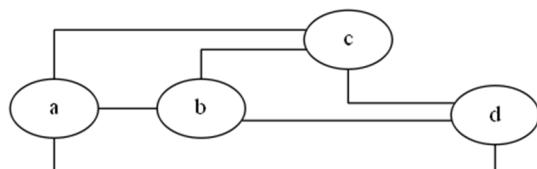
- **Line** - Edges are drawn as straight lines between nodes



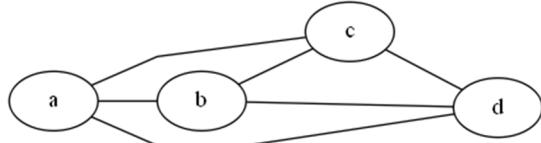
- **None** - Edges (and edge labels) are not drawn between nodes, but the relationships described by the edges affects the placement of the nodes.



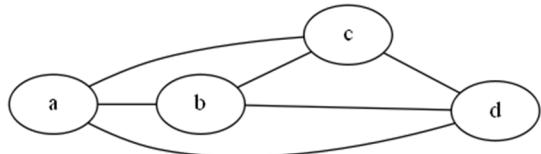
- **Ortho** - Edges are drawn with 90-degree angles in the routes between nodes.



- **Polyline** - Edges are drawn with straight lines and angular bends in the routes between nodes.

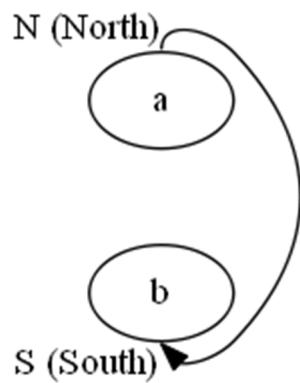


- **Spline** - Edges are drawn with straight and free-flowing (curvy) lines in the routes between nodes.



Ports

A port name can be combined with the node name to indicate where to attach an edge to the node. Graphviz has built-in port names N, S, E, W, NE, NW, SE, SW, C corresponding to compass points North, South, East, West, North East, North West, South East, South West and Center respectively.



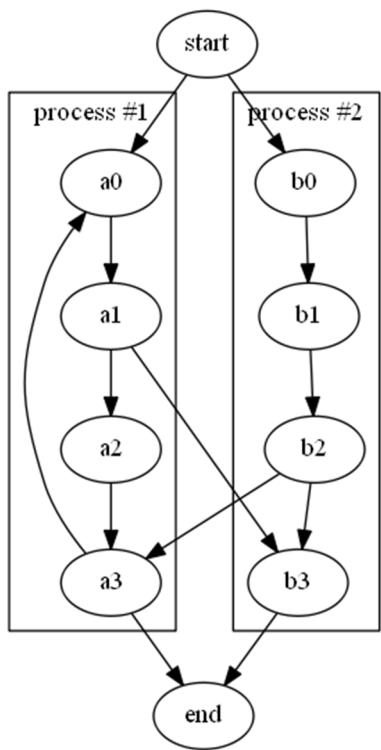
Custom ports can also be specified when using HTML labels or "record" as the node shape. This feature is explained later in this manual.

Clusters / Subgraphs

"Clusters" is a feature to draw nodes and edges in a separate rectangular layout region. Clusters exist as subgraphs of a parent graph internal to Graphviz.

Only the "dot", "fdp", "neato" and "osage" layout engines (described in the next section) draw clusters.

In the example that follows, the rectangles labeled "process #1" and "process #2" are clusters (subgraphs) within the overall graph.



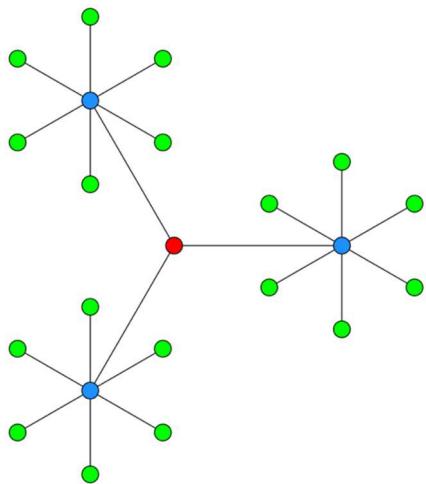
Layout Algorithms

Graphviz contains several programs for drawing graphs. Each program has specializations in how they determine how to layout the nodes and edges. Choosing a layout algorithm to use is sometimes a trial-and-error exercise to find which output looks the best.

A description of the layout engines available (as documented on the Graphviz homepage) are as follows:

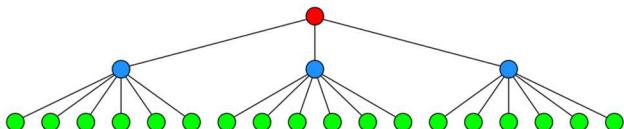
circo

circular layout, after Six and Tollis 99, Kauffman and Wiese 02. This is suitable for certain diagrams of multiple cyclic structures, such as certain telecommunications networks.



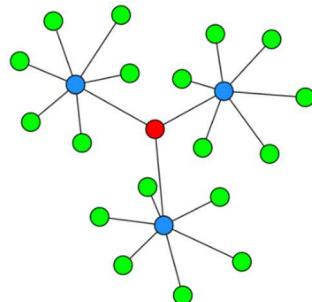
dot

"hierarchical" or layered drawings of directed graphs. This is the default tool to use if you want to have some control regarding the direction of how the graph is drawn.



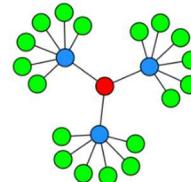
fdp

"spring model" layouts like those of neato but does this by reducing forces rather than working with energy.



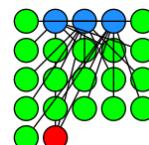
neato

"spring model" layouts. This is the default tool to use if the graph is not too large (about 100 nodes) and you do not know anything else about it. Neato attempts to minimize a global energy function, which is equivalent to statistical multi-dimensional scaling.



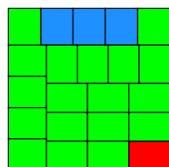
osage

The osage layout algorithm is for large undirected graphs with multiple subgraphs. It separates the graph into "levels" (clusters) and lays out each level in a rectangle. The rectangles are then packed together. Within each rectangle, the subgraph/cluster is laid out.



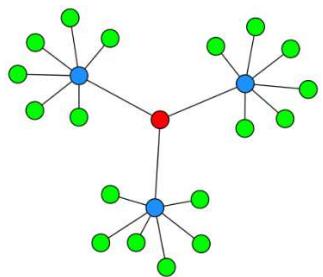
patchwork

The patchwork layout engine draws the graph as a squarified treemap. The clusters on the graph are used to create the tree.



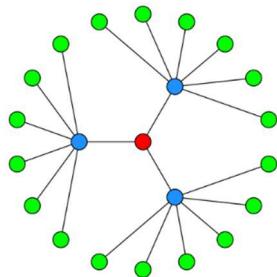
sfdp

Multiscale version of *fdp* for the layout of large graphs.



twopi

Radial layouts, after Graham Wills 97. Nodes are placed on concentric circles depending their distance from a given root node.

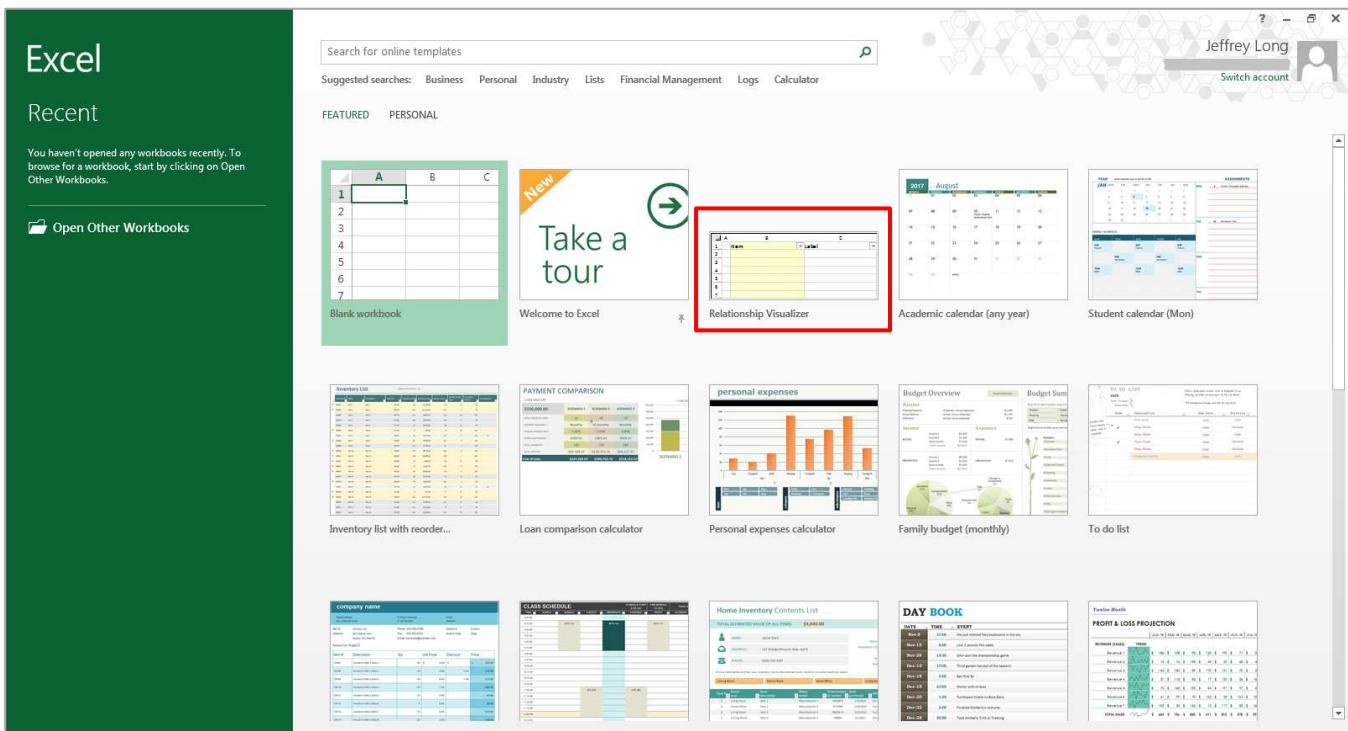


CREATING GRAPHS

If you have made it this far in this manual, Congratulations. We are past all the software installation steps, graph theory discussions, and definitions. It is time to have some fun and create some simple diagrams.

Create a New Workbook from the Relationship Visualizer Template

The first action is to launch Excel. When Excel starts, it will suggest sample spreadsheets you can create. This will contain the Relationship Visualizer template which you saved as a template as part of the installation steps. Select this template to create a new workbook. (Note: If you do not see it in the FEATURED list, click on PERSONAL to see your individual list.)



Save the Workbook as a Macro-Enabled Workbook

The workbook will appear as shown below.

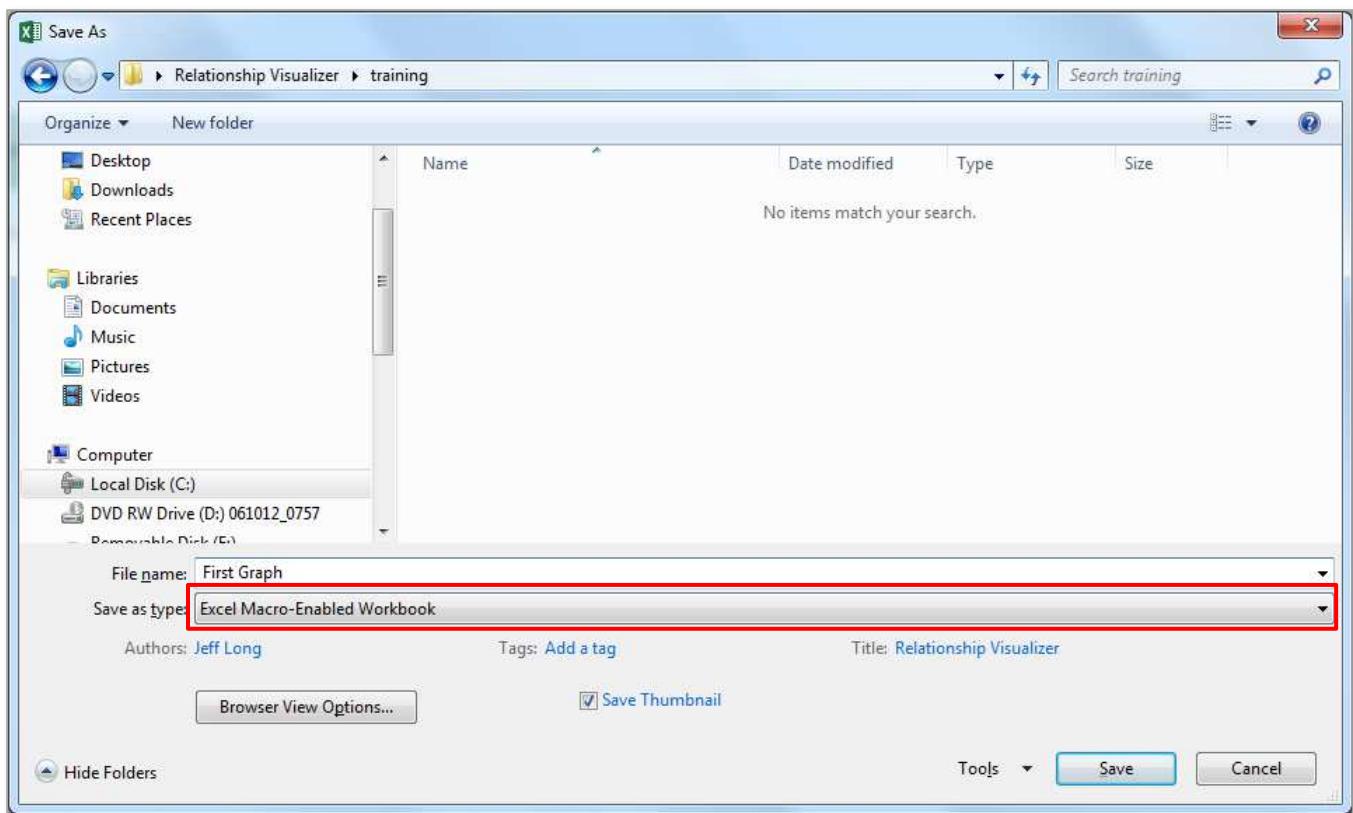
Item	Label	Related Item	Style Name	Attributes
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				
37				
38				

At the bottom of the ribbon, the tabs are: data, graph, styles, style designer, about..., and a plus sign.

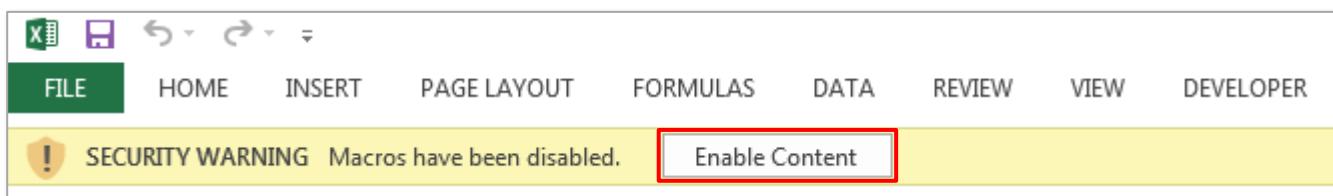
Perform a "**FILE -> Save As**" action. Choose a directory where you would like to save the file and change the file name from "Relationship Visualizer1" to something meaningful to you.

The most important step is to set the "Save as type:" dropdown list item as "**Excel Macro-Enabled Workbook**". You will not be able to run the macros that create the visualizations unless the workbook is "macro-enabled".

Excel to Graphviz for Win OS



You may or may not receive a security warning that the workbook contains macros. If you get such a warning, click the "Enable Content" button to acknowledge the risk and enable the macros.

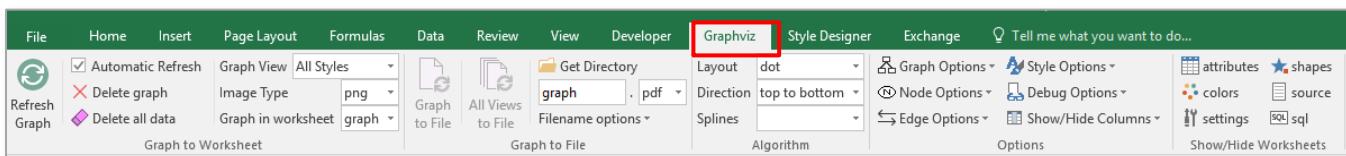


The Relationship Visualizer provides macros through five Excel Fluent UI ribbon tabs named 'Graphviz', 'Style Designer', 'SQL', and 'Source'. The appropriate ribbon tab will activate as you change worksheets.



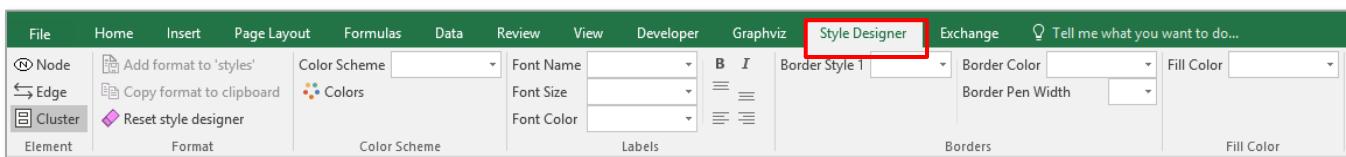
Note: There is a known bug in Excel pertaining to custom ribbons. It occurs after a "File->Save As" action changes the Excel workbook file name. The ribbon holds a reference to the original file name and breaks the ability to programmatically switch the tab focus. You can manually switch tabs as you move between worksheets, or you can close the file, and reopen it to have the ribbon tabs automatically change according to worksheet selections.

The 'Graphviz' tab contains the action buttons to create the graphs, and many Graphviz option choices to control how the graph will look.



This tab is explained in more detail in [The "Graphviz" Ribbon Tab](#) beginning on page 56.

The 'Style Designer' ribbon tab contains the action buttons and settings to create style attribute strings for nodes, edges, and clusters. The 'style designer' worksheet uses this tab exclusively.



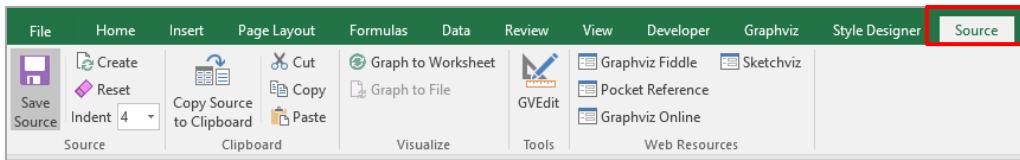
The 'Style Designer' ribbon tab is explained in more detail in [Using the 'style designer' Worksheet](#) on page 78.

The 'SQL' ribbon tab contains the action buttons to run Excel SQL statements. The 'sql' worksheet uses this tab exclusively.



The 'SQL' ribbon tab is explained in more detail in [Using SQL to Import Data from other Excel Spreadsheets](#) on page 200.

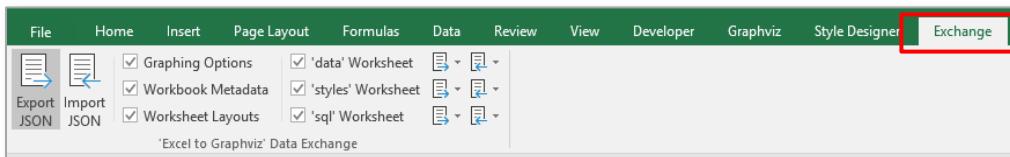
The 'Source' ribbon tab contains the action buttons to creating, viewing, and saving Graphviz source code from the information in the 'data' worksheet. The 'source' worksheet uses this tab exclusively.



The 'Source' ribbon tab is explained in more detail in [Working with the DOT Language](#) on page 209.

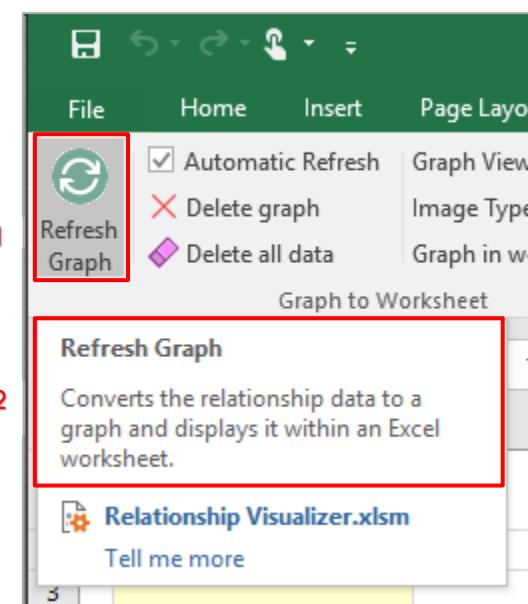
Excel to Graphviz for Win OS

The Exchange ribbon tab contains the action buttons to export or import the data contents of a Relationship Visualizer spreadsheet to a text file in JSON format. There is no worksheet associated with this tab as it accesses data from multiple worksheets.



The 'Exchange' ribbon tab is explained in more detail in [The "Exchange" Ribbon](#) on page 156.

All ribbon controls contain tool tips which explain their purpose, such as example below for the "Refresh Graph" button on the 'Graphviz' ribbon tab. Simply pause the mouse over the control (1) to make the tool tips appear (2).



Creating Your First Graph

For our first example, we will make the simplest graph possible with the tool. This graph will be a directed graph from a node named 'a' to a node named 'b'. To do so we must enter the edge information into the "data" worksheet. The "data" worksheet has 10 columns:

1. Column A - The "**Indicator**" column. This column allows you to specify a flag to treat the row as a comment. To do so enter a '#' hash tag character in the column. The row will turn green, and no data in this row will be included in the graph. If errors are detected in your data a '!' exclamation mark character will appear in the column, the row will turn red, and an error message will be displayed in the "Messages" column (column H).
2. Column B - The "**Item**" column. This column serves two purposes. For nodes, it is a unique identifier of the node. For edges, it is the unique identifier of the "from" node in a ("from", "to") node pairing.
3. Column C - The "**Label**" column. This column is optional. When specified for nodes, the value is placed inside the shape; for edges, the value is placed near the spline.
4. Column D - The "**Outside Label**" column. This column is also optional. When specified for nodes, the value is placed outside the shape, typically above and to the left of the shape; for edges, the value is placed away from the spline. If neither a "Label" or "Outside Label" is specified then the graph will default to showing the "Item" value as the inside label of nodes, and no data for edges.
5. Column E - The "**Tail Label**" column. This column is only used when specifying a relationship. It is the text label to be placed near the tail of an edge.
6. Column F - The "**Head Label**" column. This column is only used when specifying a relationship. It is the text label to be placed near the head of an edge.
7. Column G - The "**Related Item**" column. This column is only used when specifying a relationship. It is the unique identifier of the "to" node in a ("from", "to") node pairing.
8. Column H - The "**Style**" column. Optional information in this column indicates what kind of data is in the row and relates it back to a style definition in the "styles" worksheet which controls how the node or element should be graphed. For now, we will only use the styles of "edge" and "node" in our graphs. Later chapters will explain how you can create additional styles to introduce shapes, colors, and images into your graphs.
9. Column I - The "**Attributes**" column. Optional information in this column provides a means to add extra elements of style which will only apply to a single row. For example, you can place style attributes in this column which would allow you to illustrate the route from Michigan to California as described in the introduction to this manual.
10. Column J - The "**Messages**" column. When the graphing macros run they check for common mistakes in the data, such as specifying only one node for an edge. When mistakes are found, they are reported in this column. In addition, an exclamation mark '!' is placed in column 1, and the row is highlighted in red to draw your attention to the error.

The columns for "Outside Label", "Tail Label", "Head Label" and "Attributes", and Messages (columns D, E, F, H, I, and J) are hidden by default, since they are less frequently used. Display of these columns can be quickly

Excel to Graphviz for Win OS

toggled to be visible by selecting the column name on the list of columns in the dropdown list beneath the "Show/Hide Columns" button.

The screenshot shows a Microsoft Excel window titled "Relationship Visualizer.xlsx - Excel". The ribbon is active on the "Graphviz" tab. A dropdown menu labeled "Show/Hide Columns" is open on the right side of the ribbon, with a red box highlighting it. The dropdown list contains several items with checkmarks: "Comment Indicator", "Item", "Tail Label", "Label", "Related Item", "Style Name", "Attributes", and "Messages". Below the dropdown, there is a table with columns labeled A through K. Rows 1 and 2 have "Item" and "Related Item" values entered. Row 3 has "Item" and "Related Item" values entered. Row 4 has "Item" and "Related Item" values entered. Row 5 has "Item" and "Related Item" values entered. Row 6 has "Item" and "Related Item" values entered. Row 7 has "Item" and "Related Item" values entered. Row 8 has "Item" and "Related Item" values entered. Row 9 has "Item" and "Related Item" values entered. The "Style Name" and "Attributes" columns are shaded blue.

The simplest way to draw a graph is to place values in the "Item" and the "Related Item" columns. If the 'Automatic Refresh' checkbox is checked the graph will draw as data is entered into each cell. If unchecked, then press the "Refresh Graph" button.

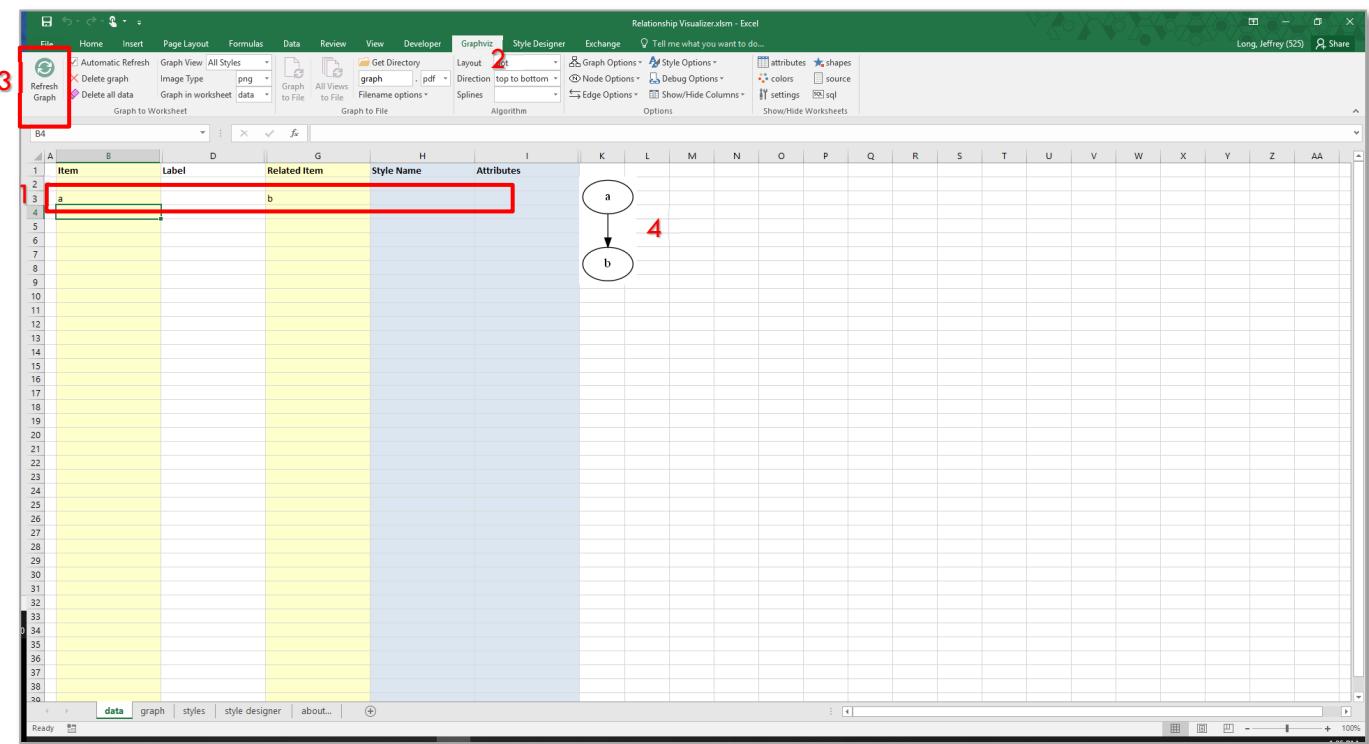
For our first graph, we will draw an 'a' is related to 'b' relationship.

1. In row 3 type 'a' in the "Item" column, and 'b' in the "Related Item" column.
2. Click on the "Graphviz" ribbon tab to activate it (if it is not the current active tab)
3. Press the **Refresh Graph** button



4. See the result beside the data

The results should resemble the following example:



Congratulations, you have created your first graph!

The next section will discuss the concepts of creating graphs in Excel. You only need to use the 'Refresh Graph' button of the 'Graphviz' ribbon tab for these tutorials. We explain 'Graphviz' ribbon tab details after the graphing concepts are mastered.

Constructing Simple Graphs

Next, lets expand upon the graph we just created to have additional relationships. Assume that:

- 'a' is related to 'b' (already drawn)
- 'b' is related to 'c'
- 'c' is related to 'a'

The Excel data appears as shown on rows 3-5. Press the 'Refresh Graph' button, and the Excel worksheet now looks like:

A	B	D	G	K	L
1	Item	Label	Related Item		
2					
3	a		b		
4	b		c		
5	c		a		
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					

```
strict digraph "main"
{
    layout="dot";
    rankdir="TB";

    "a" -> "b";
    "b" -> "c";
    "c" -> "a";
}
```

Graphviz Source

Now, let us add data into the "Label" column to label the relationships. Fill in Column C as shown below. Press the 'Refresh Graph' button, and the Excel worksheet now looks like:

A	B	D	G	K	L	M	N
1	Item	Label	Related Item				
2							
3	a	is related to	b				
4	b	is related to	c				
5	c	is related to	a				
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

```

graph TD
    a((a)) --> a
    a --> b((b))
    b --> c((c))
    c --> a
  
```

```

strict digraph "main"
{
    layout="dot";
    rankdir="TB";

    "a" -> "b"[ label="is related to" ];
    "b" -> "c"[ label="is related to" ];
    "c" -> "a"[ label="is related to" ];
}
  
```

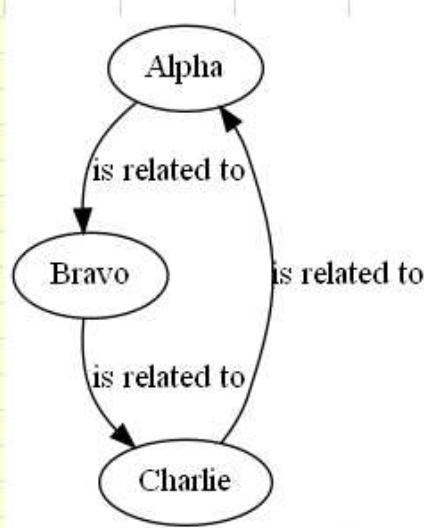
Graphviz Source

Excel to Graphviz for Win OS

The graph is how we want to see it, but the nodes need to be labeled. We do not want to change all our edges; however, we would like to replace 'a' with 'Alpha', 'b' with 'Bravo', and 'c' with 'Charlie'. The Relationship Visualizer assumes that when there is information in the "Item" column, but not in the "Related Item" column that the data corresponds to a node.

To label the nodes we will add 3 node definitions to the "data worksheet (rows 6, 7, 8) and press the 'Refresh Graph' button. The Excel worksheet now looks like:

A	B	D	G	K	L	M	N
1	Item	Label	Related Item				
2							
3	a	is related to	b				
4	b	is related to	c				
5	c	is related to	a				
6							
7	a	Alpha					
8	b	Bravo					
9	c	Charlie					
10							
11							
12							
13							
14							
15							
16							
17							



```
strict digraph "main"
{
    layout="dot";
    rankdir="TB";

    "a" -> "b"[ label="is related to" ];
    "b" -> "c"[ label="is related to" ];
    "c" -> "a"[ label="is related to" ];
    "a" [ label="Alpha" ];
    "b" [ label="Bravo" ];
    "c" [ label="Charlie" ];
}
```

Graphviz Source

Graphviz decides what it thinks is the best placement of the head and tail of an edge to produce a balanced graph.

Sometimes you might want to control where the edges begin or end. You can do that by specifying a port on the item id, in the same manner as a URL. Ports are identified by a colon character ":" and then a compass point (n, s, e, w, ne, nw, se, sw) or "c" for center.

If we change row 5 from the example above to say that "c" exiting from the south (c:s) is related to "a" entering at the north end of the node (a:n), the Excel data is changed slightly as shown in row 5. Press the 'Refresh Graph' button, and the Excel worksheet now looks like:

A	B	D	G	K	L	M	N
1	Item	Label	Related Item				
2							
3	a	is related to	b				
4	b	is related to	c				
5	c:s	is related to	a				
6							
7	a	Alpha					
8	b	Bravo					
9	c	Charlie					
10							
11							
12							
13							
14							
15							
16							

```
strict digraph "main"
{
    layout="dot";
    rankdir="TB";

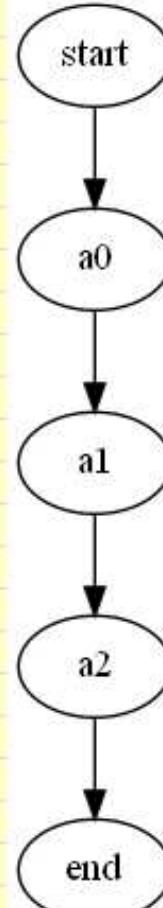
    "a" -> "b" [ label="is related to" ];
    "b" -> "c" [ label="is related to" ];
    "c":s -> "a" [ label="is related to" ];
    "a" [ label="Alpha" ];
    "b" [ label="Bravo" ];
    "c" [ label="Charlie" ];
}
```

Graphviz Source

Excel to Graphviz for Win OS

If you wish to cluster some elements of the graph you can do so by adding a row with an open brace "{" in the "Item" column above the first row of data to be placed in the group and provide a title for the cluster in the "Label" column. Next, add row with a close brace "}" in the "Item" column after the last row of data.

For example, this Excel worksheet does not have clusters.

A	B	D	G	K
1	Item	Label	Related Item	
2				
3	start		a0	
4	a0		a1	
5	a1		a2	
6	a2		end	
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				

```
strict digraph "main"
{
    layout="dot";
    rankdir="TB";

    "start" -> "a0";
    "a0" -> "a1";
    "a1" -> "a2";
    "a2" -> "end";
}
```

Graphviz Source

To cluster nodes a_0 , a_1 , and a_2 , calling the cluster "process #1" the worksheet is revised to add an open brace { with the label "process #1" on row 3, and a close brace } on rows 6 as follows.

Press the "Refresh Graph" button, and the Excel worksheet now looks like:

A	B	D	G	K	L
1	Item	Label	Related Item		
2					
3	start				
4		process #1	a0		
5	a0		a1		
6	a1		a2		
7			end		
8	a2				
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					

```

graph TD
    start((start)) --> subgraph process1 [process #1]
        process1 --> a0((a0))
        a0 --> a1((a1))
        a1 --> a2((a2))
        a2 --> end((end))
    end
  
```

```
strict digraph "main"
{
    layout="dot";
    rankdir="TB";

    "start" -> "a0";
    subgraph "cluster_1" {  label="process #1"
        "a0" -> "a1";
        "a1" -> "a2";
    }
    "a2" -> "end";
}
```

Graphviz Source

Graphviz permits clusters within clusters. Let us extend the example by adding an additional set of braces to cluster the relationship between a1 and a2. We will insert a new row 5 placing an open brace { in the "Item" column with the Label column set to "process #2", and a new row 7 with a close brace } in the "Item" column.

Press the "Refresh Graph" button, and the Excel worksheet now looks like:

A	B	D	G	K	L	M
1	Item	Label	Related Item			
2						
3	start		a0			
4	{	process #1				
5	a0		a1			
6	{	process #2				
7	a1		a2			
8	}					
9	}					
10	a2		end			
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						

```

graph TD
    start((start)) --> p1[process #1]
    p1 --> a0((a0))
    a0 --> p2[process #2]
    p2 --> a1((a1))
    a1 --> a2((a2))
    a2 --> end((end))
  
```

```
strict digraph "main"
{
    layout="dot";
    rankdir="TB";

    "start" -> "a0";
    subgraph "cluster_1" { label="process #1"
        "a0" -> "a1";
        subgraph "cluster_2" { label="process #2"
            "a1" -> "a2";
        }
    }
    "a2" -> "end";
}
```

Graphviz Source

Graphviz does not limit the number of clusters you can have. In this example, we have added rows 10-14 to insert an additional cluster labeled "process #3".

Press the "Refresh Graph" button, and the Excel worksheet now looks like:

A	B	D	G	K	L	M	N
1	Item	Label	Related Item				
2							
3	start		a0				
4	{	process #1					
5	a0		a1				
6	{	process #2	a2				
7	a1						
8	}						
9	}						
10	a2		end				
11							
12	start		b0				
13	{	process #3					
14	b0		b1				
15	}						
16	b1		end				
17							
18							
19							
20							
21							
22							
23							
24							
25							

```

graph TD
    start((start)) --> a0((a0))
    a0 --> a1((a1))
    a1 --> a2((a2))
    a2 --> end((end))

    start --> b0((b0))
    b0 --> b1((b1))
    b1 --> end

    b0 --> end
  
```

```
strict digraph "main"
{
    layout="dot";
    rankdir="TB";

    "start" -> "a0";
    subgraph "cluster_1" { label="process #1"
        "a0" -> "a1";
        subgraph "cluster_2" { label="process #2"
            "a1" -> "a2";
        }
    }
    "a2" -> "end";
    "start" -> "b0";
    subgraph "cluster_3" { label="process #3"
        "b0" -> "b1";
    }
    "b1" -> "end";
}
```

Graphviz Source

What is important to note is that you must ensure that you have an equal number of open braces as you do close braces. **If you have a mismatch between the number of open and close braces, then Graphviz will not draw the graph.**

Another feature of the Relationship Visualizer is the ability to specify a comma-separated list of item names and have a relationship created for each item. For example, we can say that Mr. Brady is the father of Greg, Peter, and Bobby on one row as follows:

A	B	D	G	K	L	M	N	O
1	Item	Label	Related Item					
2								
3	Mr. Brady	Father of	Greg, Peter, Bobby					
4								
5								
6								
7								
8								
9								
10								
11								

```
strict digraph "main"
{
    layout="dot";
    rankdir="TB";

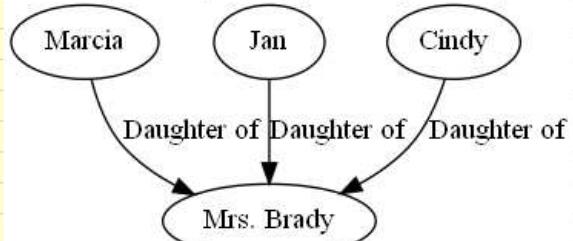
    "Mr. Brady" -> "Greg"[ label="Father of" ];
    "Mr. Brady" -> "Peter"[ label="Father of" ];
    "Mr. Brady" -> "Bobby"[ label="Father of" ];
}
```

Graphviz Source

Excel to Graphviz for Win OS

The comma-separated list can also appear in the "Item" column, such as:

A	B	D	G	K	L	M	N	O	P
1	Item	Label	Related Item						
2									
3	Marcia, Jan, Cindy	Daughter of	Mrs. Brady						
4									
5									
6									
7									
8									
9									
10									
11									

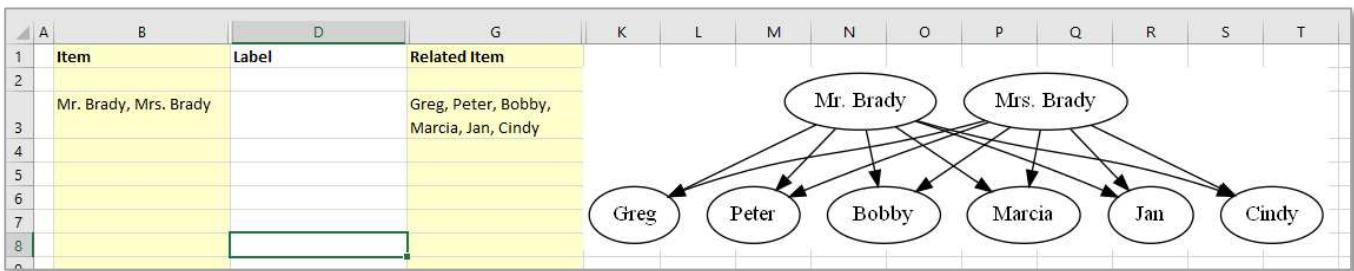


```
strict digraph "main"
{
    layout="dot";
    rankdir="TB";

    "Marcia" -> "Mrs. Brady"[ label="Daughter of" ];
    "Jan" -> "Mrs. Brady"[ label="Daughter of" ];
    "Cindy" -> "Mrs. Brady"[ label="Daughter of" ];
}
```

Graphviz Source

Or a comma-separated list can be used in both the "Item", and the "Related Item" column such as the parental relationship below:



```

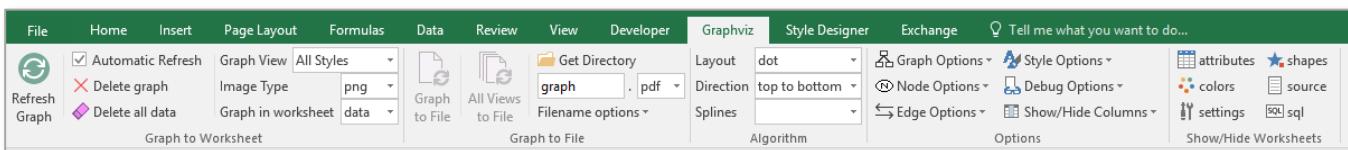
strict digraph "main"
{
    layout="dot";
    rankdir="TB";

    "Mr. Brady" -> "Greg";
    "Mr. Brady" -> "Peter";
    "Mr. Brady" -> "Bobby";
    "Mr. Brady" -> "Marcia";
    "Mr. Brady" -> "Jan";
    "Mr. Brady" -> "Cindy";
    "Mrs. Brady" -> "Greg";
    "Mrs. Brady" -> "Peter";
    "Mrs. Brady" -> "Bobby";
    "Mrs. Brady" -> "Marcia";
    "Mrs. Brady" -> "Jan";
    "Mrs. Brady" -> "Cindy";
}
  
```

Graphviz Source

The "Graphviz" Ribbon Tab

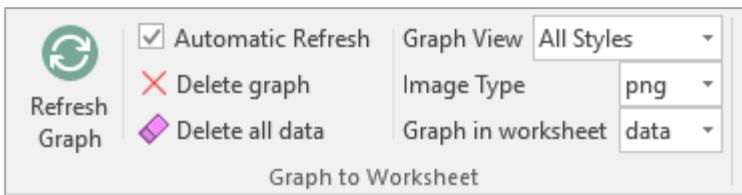
Now that you understand the basics to create graphs, let us look at the features contained in the 'Graphviz' ribbon tab. The 'Graphviz' ribbon tab is activated whenever the 'data' 'graph', 'styles', 'settings' or 'about...' worksheet is activated. It appears as follows:



It contains the following major groups:

- [Graph to Worksheet](#)
- [Graph to File](#)
- [Algorithm](#)
- [Options](#)
- [Show/Hide Worksheets](#)

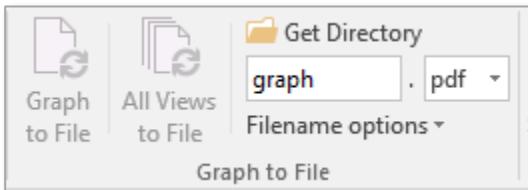
Graph to Worksheet



Label	Control Type	Description
Refresh Graph	Button	The action button that causes the Excel data to be graphed by Graphviz and then displayed within the Excel workbook.
Automatic Refresh	Checkbox	When checked, keystrokes are monitored and as cell changes are detected the graph is automatically refreshed (also requires that 'Graph in worksheet' is set to 'data').
Delete Graph	Button	Clicking on this button will delete the graph from the worksheet. This is useful when adding rows as new rows will stretch the image. You may also find you want to delete the image before saving the file to reduce the file size.
Delete all data	Button	Resets the 'data' worksheet to blank cells, and deletes any graphs if present.
Graph View	Dropdown list	The name of the column in the 'styles' worksheet which controls which set of Yes/No values to use when creating the diagrams. This dropdown list is explained in more detail in the section Creating Views beginning on page 145

Label	Control Type	Description
Image Type	Dropdown list	<p>Image format to use when displaying the graph on the "data" or "graph" worksheet of the Relationship Visualizer.</p> <p>Choices:</p> <ul style="list-style-type: none"> • bmp - Microsoft Windows Bitmap format • gif - Graphics Interchange Format • jpg - Joint Photographic Experts Group format • png - Portable Network Graphics format
Graph in worksheet	Dropdown list	<p>The worksheet in the current workbook where the graph should be displayed</p> <p>Choices:</p> <ul style="list-style-type: none"> • data - The graph is displayed in the 'data' worksheet to the left of the data • graph - The graph is displayed in the 'graph' worksheet, and the 'graph' worksheet is activated. <p>This setting is useful for large graphs as it allows you to use Excel's magnification Zoom-In/Zoom-out feature. It is also useful when you want to flip back and forth between the data and the graph to correct errors in the data.</p>

Graph to File



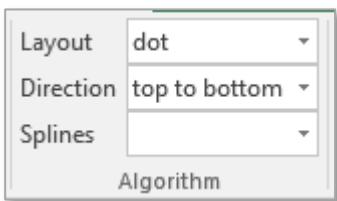
A tutorial on how to use these ribbon options is contained in the section **Publishing Graphs** that begins on page 152.

Label	Control Type	Description
Graph to File	Button	<p>The action button that causes the Excel data to be graphed by Graphviz and then written to a file.</p> <p><i>Note that this button is disabled until a valid output directory is specified.</i></p>

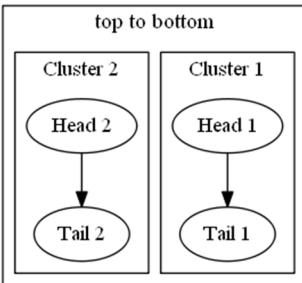
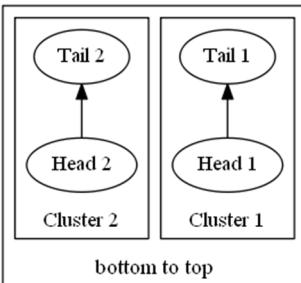
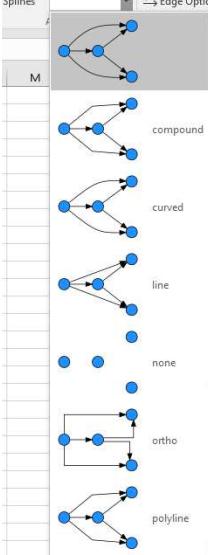
Label	Control Type	Description
All views to File	Button	<p>The action button that causes the Excel data to be graphed by Graphviz and then written to a file repeatedly for every view defined in the 'Styles' worksheet.</p> <p>NOTE: This button is disabled until a valid output directory is specified.</p>
Get Directory	Button	<p>Brings up the Directory Selection dialog and stores/displays the directory where the files should be written to. Once a directory is selected the directory path replaces the "Get Directory" button label.</p>
File Prefix	Edit box	<p>Base portion of the file name. For example: 'Graph'.</p> <p>You may also build a file name using the following character strings in the file prefix to insert run-time values into the file name.</p> <ul style="list-style-type: none"> • %D - Current date • %T - Current time • %E - Graphviz layout engine • %S - Splines • %V - View name • %W - Worksheet name <p>NOTE: You must check the appropriate options in the Filename options' dropdown list for the substitutions to occur.</p>
File Format	Dropdown List	<p>File format of the output file.</p> <p>Choices:</p> <ul style="list-style-type: none"> • bmp - Microsoft Windows Bitmap format • gif - Graphics Interchange Format • jpg - Joint Photographic Experts Group format • pdf - Portable Document Format • png - Portable Network Graphics format • ps - Postscript format • svg - Scalable Vector Graphics format • tiff - Tagged Image File Format
Filename options ▾	Dropdown List	A list of options which can be checked which will cause run-time information to be appended or omitted from the file name.

Label	Control Type	Description
Add date/time to the filename	Check box	Option to add a date and time to the file name. <ul style="list-style-type: none"> • checked, add the date and time • unchecked, omit the date and time
Add Layout/Splines to the filename	Check box	Option to add the layout engine and spline type to the file name. <ul style="list-style-type: none"> • checked, add the options • unchecked, omit the options

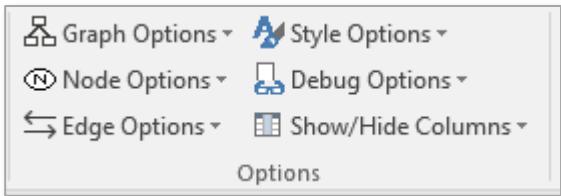
Algorithm



Label	Control Type	Description
Layout	Dropdown list	<p>Specifies which Graphviz layout program to use when drawing the graph.</p> <p>Choices:</p> <ul style="list-style-type: none"> • circo • dot • fdp • neato • osage • patchwork • sfdp • twopi <p>See Layout Algorithms on page 32 more information on the algorithms.</p>
Direction	Dropdown list	<p>This setting controls the way that shapes are placed when the dot layout engine is used. This option will be hidden from the ribbon if the graphing engine is set to anything other than dot.</p> <p>Choices:</p> <ul style="list-style-type: none"> • top to bottom • bottom to top • left to right • right to left

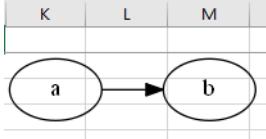
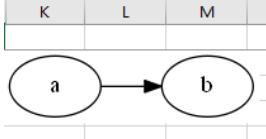
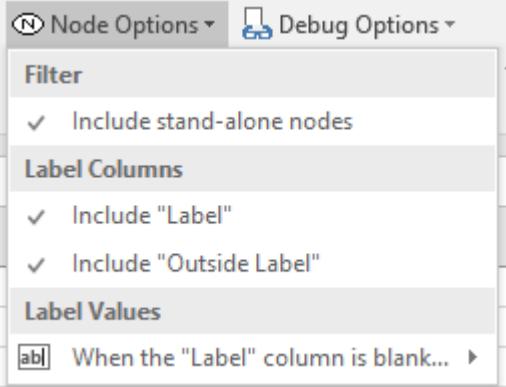
Label	Control Type	Description
		 <p>top to bottom</p>
		 <p>bottom to top</p>
Splines	Dropdown list	<p>Specifies how the edges between nodes should be drawn. See the examples in the Splines section beginning on page 30.</p> <p>Choices:</p> <ul style="list-style-type: none"> • curved • line • none • ortho • polyline • spline
		

Options

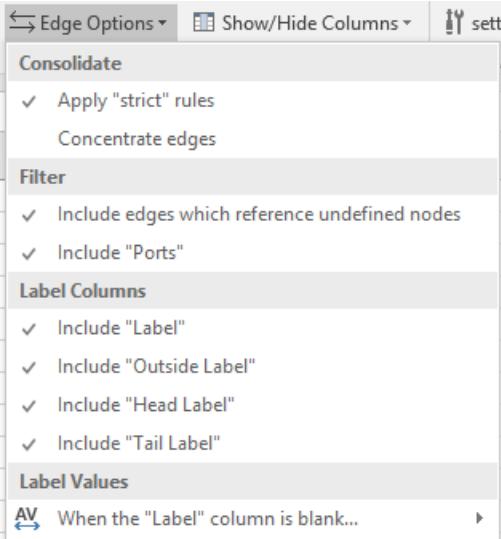
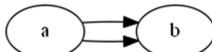


Label	Control Type	Description
Graph Options	Dropdown List	Optional attributes which can be checked for inclusion in the Graphviz source. These attributes have graph-level scope.
Graph Options	Section Heading	
Style Options	Section Heading	
Clusters	Section Heading	
Allow edges between clusters		
Rank ignoring clusters		
Drawing	Section Heading	
Center drawing		
<input checked="" type="checkbox"/> Directed graph		
Force xlabel placement		
Output Order		
Rotate 90° counterclockwise		
Transparent Background		
Clusters	Section Heading	
Allow edges between clusters	Checkbox	Adds a <code>compound="true"</code> attribute when checked. If <code>compound="true"</code> , allows edges between clusters. See Clusters - Depicting a Relationship from or to a Cluster on page 171 for more information.
Rank ignoring clusters	Checkbox	Adds a <code>newrank="true"</code> attribute when checked. This choice will only be present if the layout algorithm is set to dot. See Clusters - Aligning Nodes across Clusters on page 179 for more information.
Drawing	Section Heading	

Label	Control Type	Description
Center Drawing	Checkbox	Checking this item will cause the graph to be centered in the page, assuming the graph is smaller than the page size.
Directed graph	Checkbox	<p>Specifies if the graph should be a directed graph (i.e., with arrows) or an undirected graph (i.e., no arrows). See the examples in the Undirected Graph and Directed Graph sections on page 29.</p> <p>Choices:</p> <ul style="list-style-type: none"> • Checked - Graph as a directed graph • Unchecked - Graph as an undirected graph
Force xlabel placement	Checkbox	If checked, all xlabel attributes are placed, even if there is some overlap with nodes or other labels.
Output Order	Menu	<p>A menu of mutually exclusive choices which specify the order in which nodes and edges are drawn.</p> <p>You may pick from:</p> <ul style="list-style-type: none"> • <i>Breadth First</i> - (the default if nothing is checked) outputorder="breadthfirst" is the simplest, but when the graph layout does not avoid edge-node overlap, this mode will sometimes have edges drawn over nodes and sometimes on top of nodes. • <i>Nodes First</i> - If the mode outputorder="nodesfirst" is chosen, all nodes are drawn first, followed by the edges. This guarantees an edge-node overlap will not be mistaken for an edge ending at a node. • <i>Edges First</i> - It may be desirable that all edges appear beneath nodes, even if the resulting drawing is ambiguous. This can be achieved by choosing outputorder="edgesfirst".
Rotate 90 counterclockwise	Checkbox	If checked, causes the final layout to be rotated counterclockwise by 90 degrees.
Transparent Background	Checkbox	Toggles the background color between white and transparent. Transparent backgrounds are useful if you intend to layer the graphs in an image editor or paste them into a Microsoft Word document.

Label	Control Type	Description
Choices:		
		<ul style="list-style-type: none"> • Checked - Background is transparent 
		<ul style="list-style-type: none"> • Unchecked - Graph background is white. 
		<p>Note: It is possible to set the graph background color to any valid color by specifying the <code>bcolor=</code> attribute as a graph option on the 'settings' worksheet.</p>
Node Options	Dropdown list	Choices which control which nodes are included in the Graphviz source, and how the labels should be represented.
		
Drawing	Label	Section Heading
Include stand-alone nodes	Checkbox	<p>Include or exclude nodes without relationships (i.e., island nodes).</p> <p>When using views to exclude relationship edges there may be nodes left in the diagram that are not connected to anything. This setting specifies if these</p>

Label	Control Type	Description
		<p>island nodes should be included or excluded from the diagram.</p> <p>Choices:</p> <ul style="list-style-type: none"> • Checked - retain the island nodes • Unchecked - drop the island nodes from the diagram
Label Columns	Label	Section Heading
Include "Label"	Checkbox	<p>Include or exclude Labels column data?</p> <p>Allows you to turn labels on/off in the graph.</p> <p>Choices:</p> <ul style="list-style-type: none"> • Checked - Include label column data • Unchecked - Drop the label column data from the graph
Include "Outside Label"	Checkbox	<p>Include or exclude Outside Labels column data?</p> <p>Allows you to turn outside (xlabel) labels on/off in the graph.</p> <p>Choices:</p> <ul style="list-style-type: none"> • Checked - Include outside label column data • Unchecked - Drop the outside label column data from the graph
When the "Label" column is blank...	Menu	<p>Include or exclude blank values in the Label column?</p> <p>When the "Label" column is blank on the data worksheet on a row which refers to a node it can mean two possible things. One interpretation is to remove the label from the node, as might be useful when using images to represent nodes. The other interpretation is to let the graph default to displaying the value in the "Item" column.</p> <p>Choices:</p> <ul style="list-style-type: none"> • ...use blank for the node label - use the blank label as the node's label text • ...use the node identifier as the label - show the value in the "Item" column as the label text

Label	Control Type	Description
Edge Options	Dropdown list	Choices which control how edges should be specified in the Graphviz source, and how the labels should be represented.
		
Consolidate	Section Heading	
Apply "strict" rules	Checkbox	<p>Specifies the <code>strict</code> attribute for the top-level graph.</p> <p>Describing the graph as strict forbids the creation of multi-edges, i.e., there can be at most one edge with a given tail node and head node in the directed case. For undirected graphs, there can be at most one edge connected to the same two nodes. Subsequent edge statements using the same two nodes will identify the edge with the previously defined one and apply any attributes given in the edge statement.</p> <p>Choices:</p> <ul style="list-style-type: none"> • Checked - Includes the <code>strict</code> attribute  <p>Edges have been consolidated.</p> <ul style="list-style-type: none"> • Unchecked - Omits the <code>strict</code> attribute  <p>Edges have not been consolidated.</p>

Label	Control Type	Description
Concentrate edges	Checkbox	<p>If checked, use edge concentrators.</p> <p>This merges multi-edges into a single edge and causes partially parallel edges to share part of their paths.</p> <p>This feature is only available if the layout algorithm is dot.</p> <p>Choices:</p> <ul style="list-style-type: none"> • Checked - Include the concentrate attribute

```

graph TD
    a((a)) --> b((b))
    c((c)) --> b
    b --> x((x))
    x --> y((y))
    y --> d((d))
    b --> d
    y --> d
  
```

The diagram illustrates a directed graph with nodes a, b, c, x, y, and d. Nodes a and c have arrows pointing to node b. Node b has arrows pointing to nodes x, y, and d. Node x has an arrow pointing to node y. Node y has an arrow pointing to node d. There are two parallel edges from b to d, which are merged into a single curved edge in the concentrated version.

Edges have been concentrated

- **Unchecked** - Omits the concentrate attribute

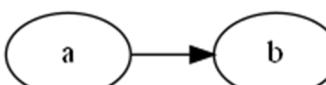
```

graph TD
    a((a)) --> b((b))
    c((c)) --> b
    b --> x((x))
    x --> y((y))
    y --> d((d))
    b --> d
    y --> d
  
```

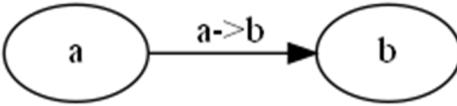
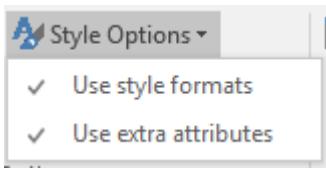
The diagram illustrates a directed graph with nodes a, b, c, x, y, and d. Nodes a and c have arrows pointing to node b. Node b has arrows pointing to nodes x, y, and d. Node x has an arrow pointing to node y. Node y has an arrow pointing to node d. Two separate edges connect b to d, and two separate edges connect y to d.

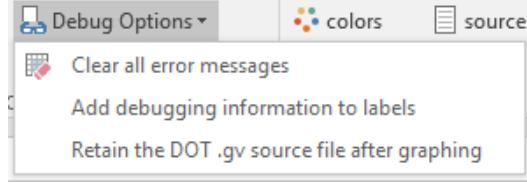
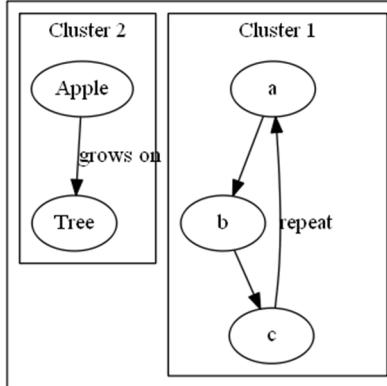
Edges are not concentrated

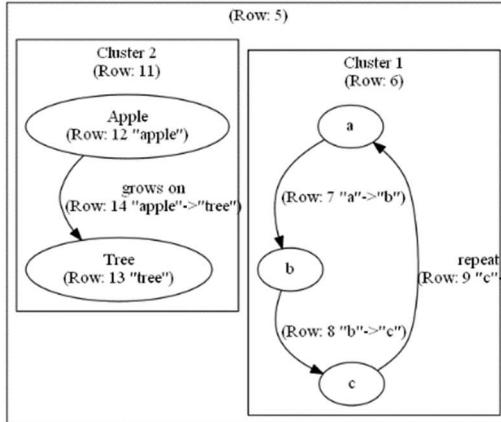
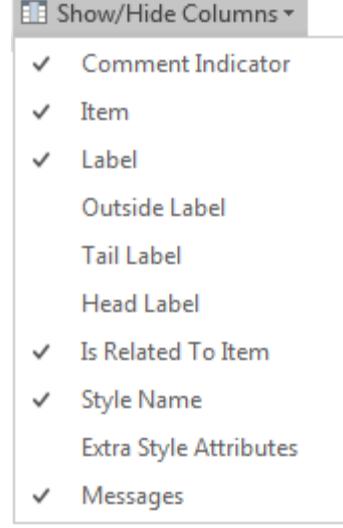
Filter	Label	Section Heading
--------	-------	-----------------

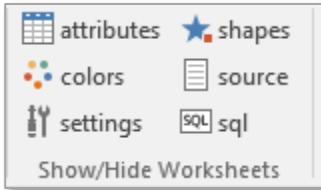
Label	Control Type	Description				
Include edges which reference undefined nodes	Checkbox	<p>Include/Exclude relationships without nodes (i.e., orphan edges).</p> <p>When using views to exclude nodes there may be un-styled nodes included in the diagram due to edge references. This setting specifies if the edges should be included or excluded from the diagram.</p> <p>Choices:</p> <ul style="list-style-type: none"> • Checked - retain edges which have references to undefined nodes • Unchecked - drop any edges which do not refer to defined nodes 				
Include Ports	Checkbox	<p>Retain/Remove port values from the nodes in an edge relationship.</p> <p>Given:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <th>Item</th> <th>Related Item</th> </tr> <tr> <td>a:n</td> <td>b:s</td> </tr> </table> <p>Choices:</p> <ul style="list-style-type: none"> • Checked - retain the ports when creating the edge syntax.  <p style="text-align: center;">$a:n \rightarrow b:s$</p> <ul style="list-style-type: none"> • Unchecked - removes the ports specified when creating the edge syntax.  <p style="text-align: center;">$a \rightarrow b$</p>	Item	Related Item	a:n	b:s
Item	Related Item					
a:n	b:s					
Label Columns	Label	Section Heading				
Include "Label"	Checkbox	<p>Include or exclude Labels column data?</p> <p>Allows you to turn edge labels on/off in the graph.</p>				

Label	Control Type	Description
		<p>Choices:</p> <ul style="list-style-type: none"> • Checked - Include Label column data • Unchecked - Omit the Label column data from the graph
Include "Outside Label"		<p>Include or exclude Outside Labels column data?</p> <p>Allows you to turn outside (xlabel) edge labels on/off in the graph.</p> <p>Choices:</p> <ul style="list-style-type: none"> • Checked - Include outside label column data • Unchecked - Omit the outside label column data from the graph
Include "Head Label"	Checkbox	<p>Include or exclude Head Labels column data?</p> <p>Allows you to turn edge head labels on/off in the graph.</p> <p>Choices:</p> <ul style="list-style-type: none"> • Checked - Include Head Label column data • Unchecked - Omit the Head Label column data from the graph
Include "Tail Label"	Checkbox	<p>Include or exclude Tail Labels column data?</p> <p>Allows you to turn edge tail labels on/off in the graph.</p> <p>Choices:</p> <ul style="list-style-type: none"> • Checked - Include Tail Label column data • Unchecked - Omit the Tail Label column data from the graph
Label Values	Label	Section Heading
When the "Label" column is blank...	Menu	<p>Include or exclude blank values in the Label column?</p> <p>When the "Label" column is blank on the data worksheet on a row which refers to an edge it can mean two possible things. One interpretation is to remove the label from the edge. The other interpretation is to let the graph default to</p>

Label	Control Type	Description						
		<p>displaying the value Graphviz assigns to the edge relationship.</p> <p>Given:</p> <table border="1"> <thead> <tr> <th>Item</th><th>Label</th><th>Related Item</th></tr> </thead> <tbody> <tr> <td>a</td><td></td><td>b</td></tr> </tbody> </table> <p>Choices:</p> <ul style="list-style-type: none"> ...the label is blank - use the blank label as the node's label text  <ul style="list-style-type: none"> ...use the edge name as the label - show the value in the "Item" column as the label text 	Item	Label	Related Item	a		b
Item	Label	Related Item						
a		b						
Style Options	Dropdown List	<p>Switches which allow you to turn predefined style definitions and ad-hoc row level attributes on or off.</p> 						
Use style formats	Dropdown list	<p>Specifies if the style attributes associated with the Style Name assigned to a node, edge, or cluster should be used when the graph is generated.</p> <p>Choices:</p> <ul style="list-style-type: none"> Checked - use the style format Unchecked - do not use the style format (i.e., use default Graphviz rendering method) 						

Label	Control Type	Description
Use extra attributes	Dropdown list	<p>Specifies if the "Attributes" style attributes on the 'data' worksheet should be included or omitted when the graph is generated.</p>
		Choices:
		<ul style="list-style-type: none"> • Checked - include the style attributes • Unchecked - do not include the style attributes
Debug Options	Dropdown List	Debug Options
		
Clear all error messages	Button	Resets the error message column
Add debugging information to labels	Checkbox	<p>Turning this option to "on" causes additional information such as the row number and item identifiers to be included in the labels of nodes, edges, and clusters.</p>
		Choices:
		<ul style="list-style-type: none"> • unchecked - Do not add information to the labels • checked - Add information to the labels
	Unchecked	

Label	Control Type	Description
		
		Checked
Retain the DOT .gv source file after graphing	<input type="checkbox"/>	<p>Specifies what should be done with the text file sent to Graphviz after the graphing step is complete when "Graph to File" is used to create the graph.</p> <p>Choices:</p> <ul style="list-style-type: none"> checked - retain the file. It will be in the same directory as the graph file with the same file name except for the file extension (which will be .gv). unchecked - delete the file
Show/Hide Columns	<input type="checkbox"/>	<p>A list of column names on the 'data' worksheet which can be displayed or hidden.</p> <p>Choices:</p> <ul style="list-style-type: none"> checked - show the column unchecked - hide the column
		

Show/Hide Worksheets

Label	Control Type	Description
Attributes	Toggle button	Specifies if the " HELP - attributes " worksheet should be visible. Choices: <ul style="list-style-type: none">• selected - shows the "HELP - attributes" worksheet• not selected - hides the "HELP - attributes" worksheet
Colors	Toggle button	Specifies if the " HELP - colors " worksheet should be visible. Choices: <ul style="list-style-type: none">• selected - shows the "HELP - colors" worksheet• not selected - hides the "HELP - colors" worksheet
Settings	Toggle-button	Specifies if the " settings " worksheet should be visible. Choices: <ul style="list-style-type: none">• selected - shows the "settings" worksheet• not selected - hides the "settings" worksheet
shapes	Toggle button	Specifies if the " HELP - shapes " worksheet should be visible. Choices: <ul style="list-style-type: none">• selected - shows the "HELP - shapes" worksheet• not selected - hides the "HELP - shapes" worksheet
source	Toggle button	Specifies if the " source " worksheet should be visible. Choices: <ul style="list-style-type: none">• selected - shows the "source" worksheet• not selected - hides the "source" worksheet
sql	Toggle button	Specifies if the " sql " worksheet should be visible. Choices: <ul style="list-style-type: none">• selected - shows the "sql" worksheet• not selected - hides the "sql" worksheet

CREATING RELATIONSHIP VISUALIZATIONS

Introduction

In the previous chapter, we saw how to use the basic features to create graphs. Now we will turn our attention to showing how to depict relationships and add meaningful elements of style to the nodes and edges that help distinguish the relationships.

The introduction section of this manual showed the power of using state border relationships to convert a list in Excel into a graph connecting the states. This was possible because there were individual rows for each state and the state it borders. This is a common way to represent data in Excel and assumes there is an implied relationship from one column to another column. The relationship is typically described in the column heading. Since many spreadsheets collect data in this manner, they can all lend themselves to graphing. For example, spreadsheets containing organization chart information, lists of software installed on computers or the completion status and due dates of business projects, or team member and their skills can all be depicted.

Readers of this manual work in different industries, so the example we will use is not industry specific. This example uses the relationships between family and friends that should resonate across all readers. For this example, I am using the fictional Cunningham family from the 1970's television program Happy Days.

'Happy Days' Example

Happy Days is an American television situation comedy that aired first-run from January 15, 1974, to September 24, 1984, on ABC. Created by Garry Marshall, the series presents an idealized vision of life in the mid-1950s to mid-1960s United States.



Set in Milwaukee, Wisconsin, the series revolves around teenager Richie Cunningham and his family: his father, Howard, who owns a hardware store; traditional homemaker and mother, Marion; younger sister Joanie; and high school dropout, biker, auto mechanic and suave ladies' man Arthur "Fonzie"/"The Fonz" Fonzarelli, who would eventually become the Cunningham's upstairs tenant. The earlier episodes revolve around Richie and his friends, Warren "Potsie" Weber and

Ralph "Hector" Malph, with Fonzie as a secondary character. However, as the series progressed, Fonzie proved to be a favorite with viewers and soon more story lines were written to reflect his growing popularity. Fonzie befriended Richie and the Cunningham family. In later seasons, other characters were introduced including Fonzie's young cousin, Charles "Chachi" Arcola, who became a love interest for Joanie Cunningham, and married Joanie in a spin-off television series "Joanie Loves Chachi".

Restating this information in terms of relationships:

- **Howard Cunningham** ("Mr. C") is the husband of Marion, Father of Richie, Father of Joanie, Landlord of Fonzie, and a Customer of Fonzie's auto repair garage
- **Marion Cunningham** ("Mrs. C") is the wife of Howard, Mother of Richie, Mother of Joanie
- **Richard Cunningham** ("Richie") is the son of Howard, son of Marion, brother of Joanie, and friend of Fonzie, Ralph, and Potsie

- **Joanie Cunningham** ("Shortcakes") is the daughter of Howard, daughter of Marion, sister of Richie, and eventually the wife of Chachi
- **Ralph Malph** ("Hector") is a friend of Richie, Potsie, and Fonzie
- **Warren Weber** ("Potsie") is a friend of Richie, Ralph, and Fonzie
- **Arthur Fonzarelli** ("Fonzie", "The Fonz") is a friend of Richie, Ralph, and Potsie; a tenant of Howard, Howard's auto mechanic, and a cousin of Chachi
- **Charles Arcola** ("Chachi") is a cousin of Fonzie, and eventually becomes Joanie's husband.

Collecting this information into an Excel spreadsheet table, it would look as follows:

A	B		C	D
1	Name	Full Name	Kind of Relationship	Relationship To
2	Howard	Howard Cunningham ("Mr. C")	Husband	Marion
3			Father	Richie
4			Father	Joanie
5			Landlord	Fonzie
6			Customer	Fonzie
7	Marion	Marion Cunningham ("Mrs. C")	Wife	Howard
8			Mother	Richie
9			Mother	Joanie
10	Richie	Richard Cunningham ("Richie")	Son	Howard
11			Son	Marion
12			Brother	Joanie
13			Friend	Fonzie
14			Friend	Ralph
15			Friend	Potsie
16	Joanie	Joanie Cunningham ("Shortcakes")	Daughter	Howard
17			Daughter	Marion
18			Sister	Richie
19			Wife	Chachi
20	Fonzie	Arthur Fonzarelli ("Fonzie", "The Fonz")	Friend	Richie
21			Friend	Ralph
22			Friend	Potsie
23			Tenant	Howard
24			Mechanic	Howard
25			Cousin	Chachi
26	Chachi	Charles Arcola ("Chachi")	Cousin	Fonzie
27			Husband	Joanie
28	Ralph	Ralph Malph	Friend	Fonzie
29			Friend	Richie
30			Friend	Potsie
31	Potsie	Warren Weber ("Potsie")	Friend	Fonzie
32			Friend	Richie
33			Friend	Ralph

Let us put this information into the Relationship Visualizer spreadsheet and observe the connections.

Step 1 - Create a new spreadsheet from the template and save it as an "**Excel Macro-enabled Workbook**" (.xslm format) as previously described.

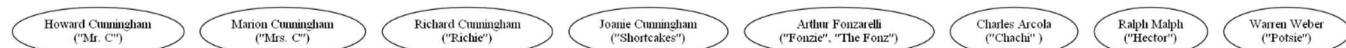
Step 2 - Take the information in Columns A, B and create nodes in the Relationship Visualizer spreadsheet. Do this by copying the "Name" values to the "Item" column, and the "Full Name" values to the "Label" column.

TIP: The Relationship Visualizer supports multi-line labels. Simply use Excel's "Alt+Enter" keystroke combination to insert a carriage return within the label text. This tip has been illustrated below by entering carriage returns to place the nicknames in parenthesis below the formal names.

Your data should look as follows:

A	B	D	G
1	Item	Label	Related Item
2			
3			
4	Howard	Howard Cunningham ("Mr. C")	
5	Marion	Marion Cunningham ("Mrs. C")	
6	Richie	Richard Cunningham ("Richie")	
7	Joanie	Joanie Cunningham ("Shortcakes")	
8	Fonzie	Arthur Fonzarelli ("Fonzie", "The Fonz")	
9	Chachi	Charles Arcola ("Chachi")	
10	Ralph	Ralph Malph ("Hector")	
11	Potsie	Warren Weber ("Potsie")	

When you press the "Refresh Graph" button, you will see a graph such as:



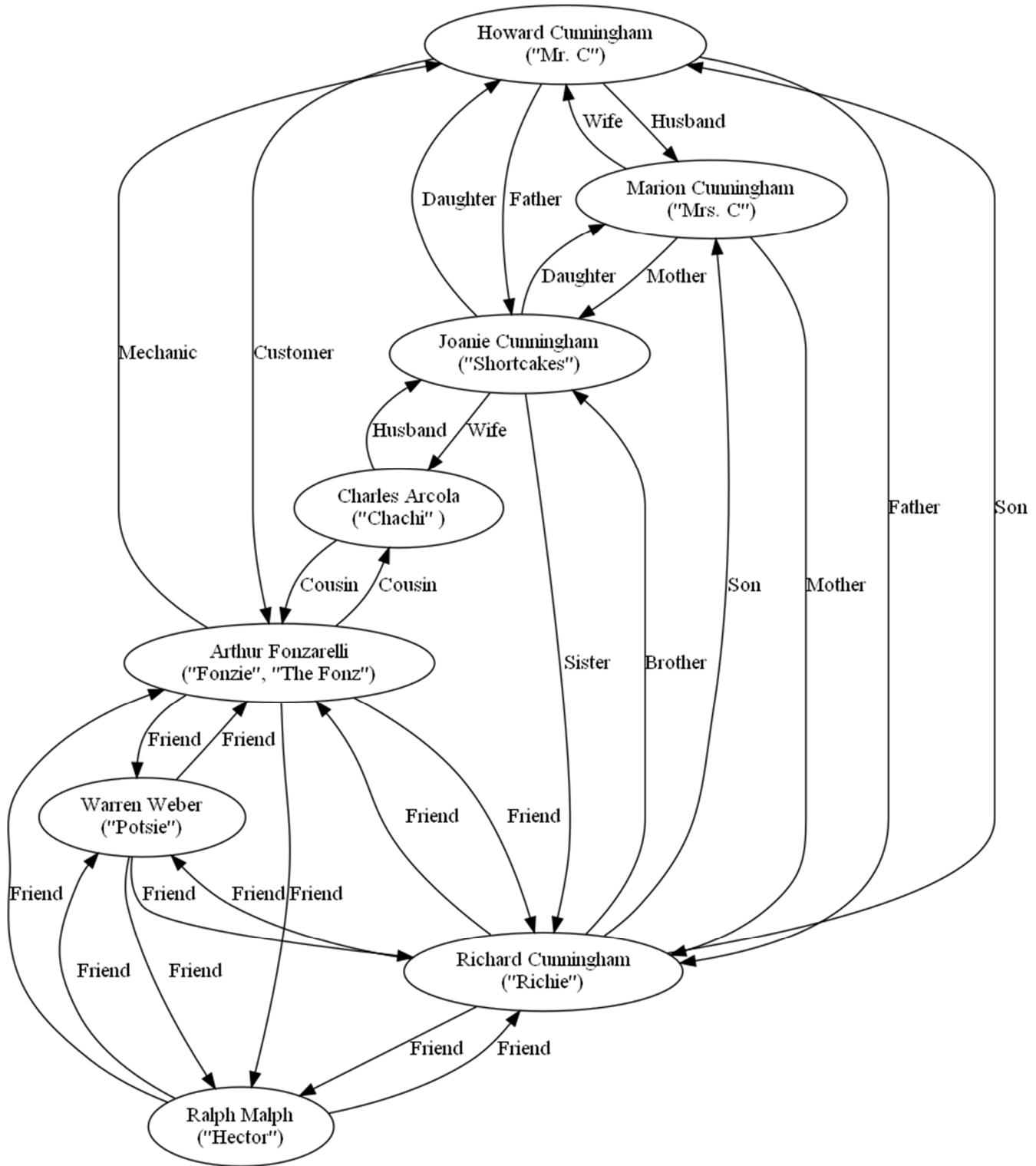
Step 3 - Create the edges. Copy into the Relationship Visualizer spreadsheet:

- the "Name" values in Column A into the "Item" column B
- the "Kind of Relationship" values in Column C into the "Label" column C
- the "Relationship To" values in Column D into the "Related Item" E

Your data should now look as follows:

A	B	D	G
1	Item	Label	Related Item
13	Howard	Husband	Marion
14	Howard	Father	Richie
15	Howard	Father	Joanie
16	Howard	Landlord	Fonzie
17	Howard	Customer	Fonzie
18	Marion	Wife	Howard
19	Marion	Mother	Richie
20	Marion	Mother	Joanie
21	Richie	Son	Howard
22	Richie	Son	Marion
23	Richie	Brother	Joanie
24	Richie	Friend	Fonzie
25	Richie	Friend	Ralph
26	Richie	Friend	Potsie
27	Joanie	Daughter	Howard
28	Joanie	Daughter	Marion
29	Joanie	Sister	Richie
30	Joanie	Wife	Chachi
31	Fonzie	Friend	Richie
32	Fonzie	Friend	Ralph
33	Fonzie	Friend	Potsie
34	Fonzie	Tenant	Howard
35	Fonzie	Mechanic	Howard
36	Fonzie	Cousin	Chachi
37	Chachi	Cousin	Fonzie
38	Chachi	Husband	Joanie
39	Ralph	Friend	Fonzie
40	Ralph	Friend	Richie
41	Ralph	Friend	Potsie
42	Potsie	Friend	Fonzie
43	Potsie	Friend	Richie
44	Potsie	Friend	Ralph

If you press "Refresh Graph" you will see a graph such as:



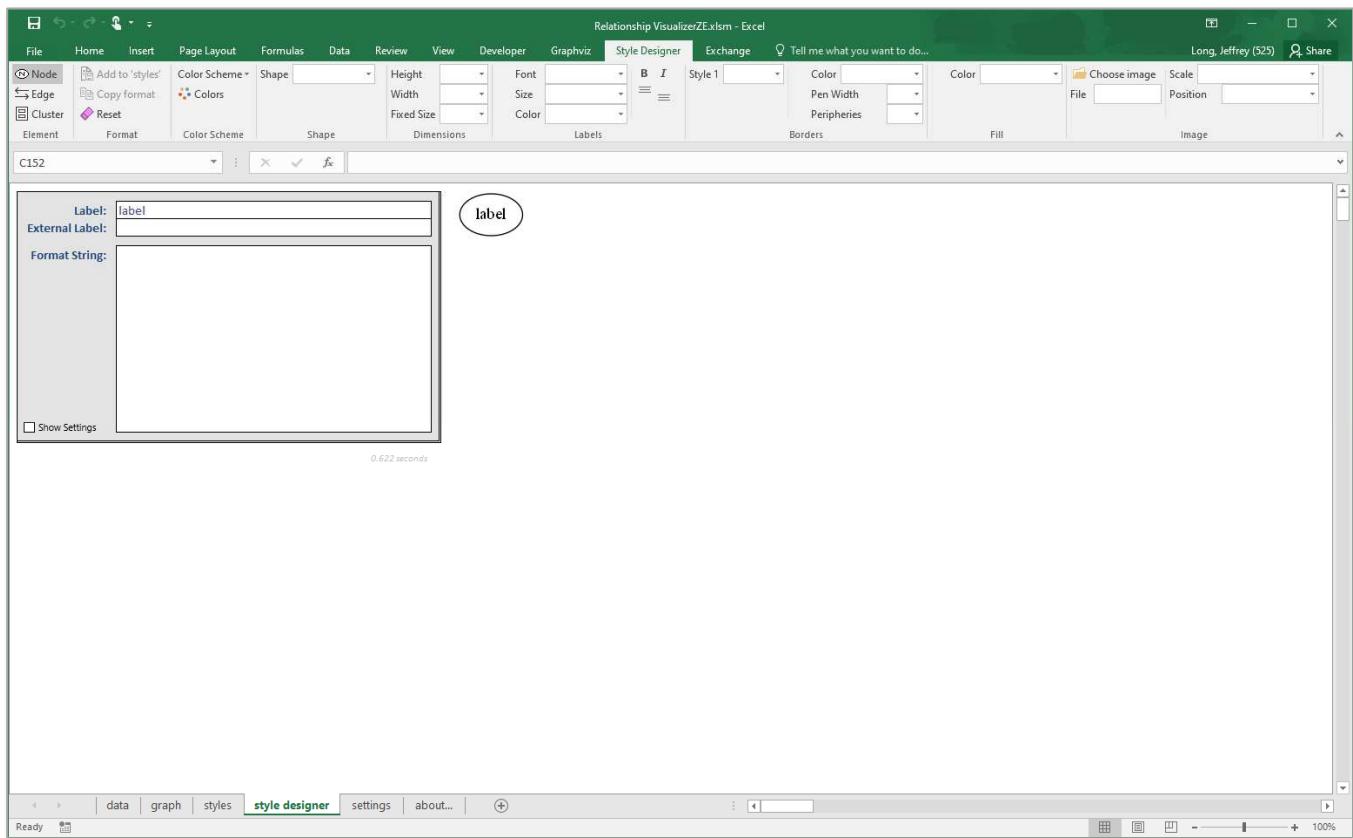
We have successfully drawn the relationships between the main Happy Days characters!

Let us not be content, we can do better. The interconnections of the relationships determine the node placement, a very powerful concept. A shortcoming is that it is hard to distinguish the types of data, and the

types of relationships. For example, it would be nice to know at a glance who are members of the Cunningham family, and who are not. It would also be nice to know at a glance what are the spouse relationships, the parent/child relationships, the friend relationships, and the business relationships. This is where the "styles" and "style designer" worksheets come into play.

Using the 'style designer' Worksheet

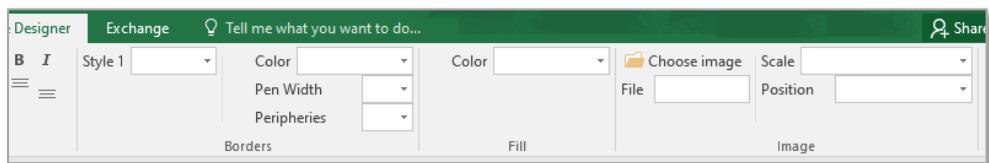
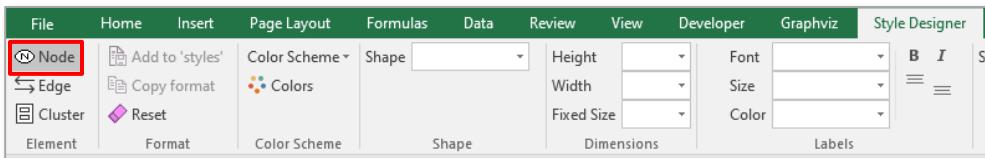
The Graphviz DOT language supports many attributes that can control the appearance of nodes and edges. The 'style designer' worksheet makes it easy for you to compose style specifications without having to know all the details of the DOT language. The 'style designer' worksheet appears as follows:



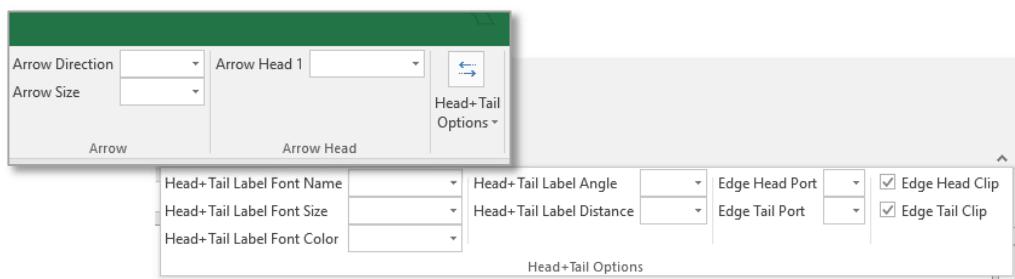
The "Style Designer" Ribbon Tab

The Style Designer ribbon tab has three dynamic design modes controlled by the 'Element' radio buttons in the upper left corner that allow you to create node styles, edge styles, and cluster styles. The ribbon controls change dynamically as you make selections.

Element = Node



Element = Edge



Element = Cluster



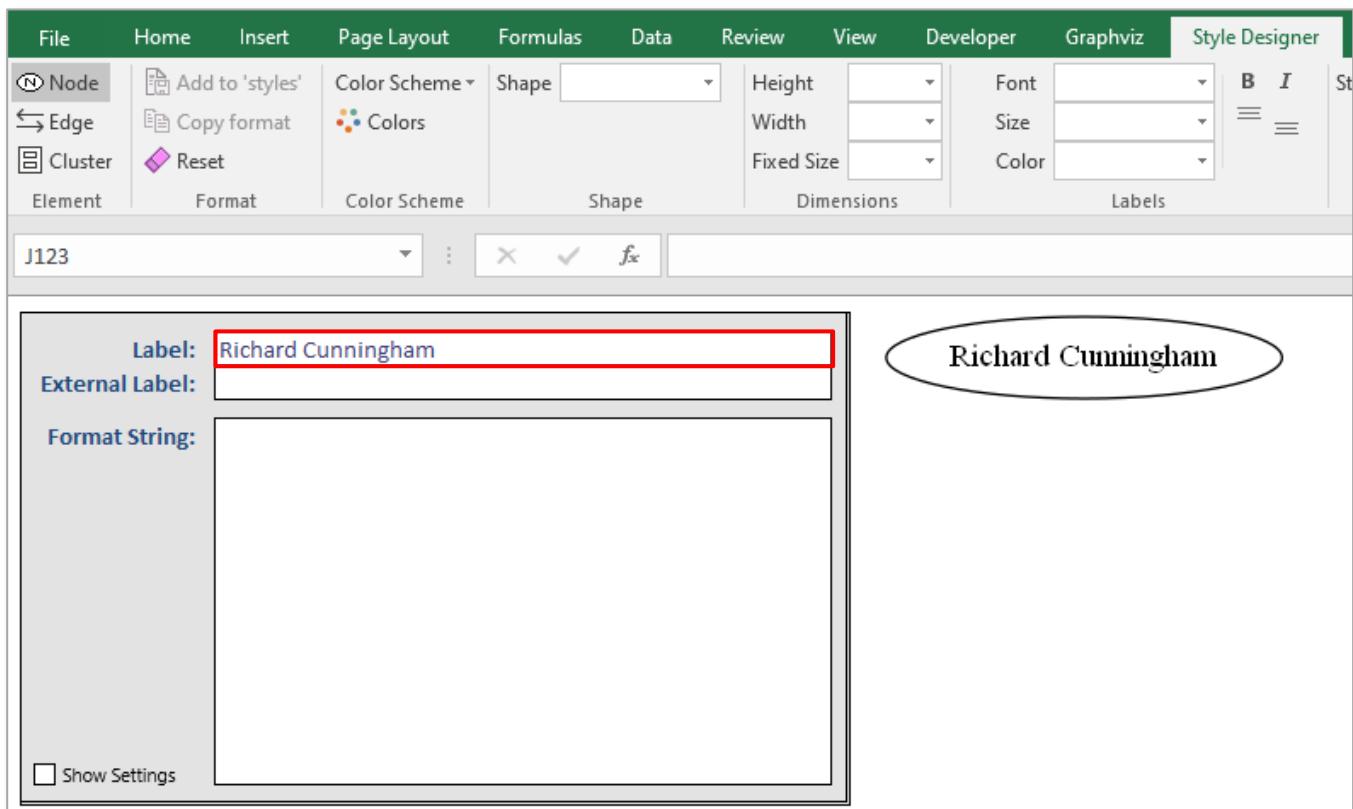
You define the styles by making selections on the "Style Designer" ribbon tab. As you make selections a format string is created, and a sample rendering of the node/edge/cluster is created using the graphing engine and splines values from the "Graphviz" ribbon tab (more on that later).

Use these elements as an aid to make your selections on the 'style designer' worksheet so that appropriate Graphviz attributes are used in context. For example, Element = Edge will not offer attributes such as "shape" which would not be appropriate for an edge.

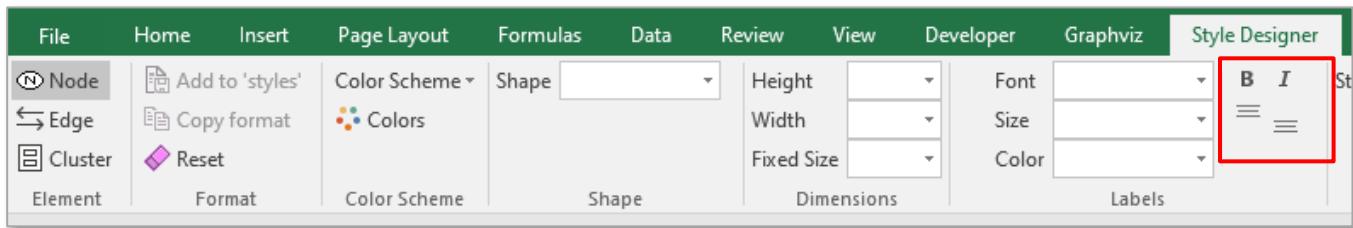
We can use the **Node Style** mode to create a unique style for members of the Cunningham family. Let's play with different values and until we obtain something visually pleasing. For the Cunningham family:

Excel to Graphviz for Win OS

Enter "Richard Cunningham" into the 'Node Label Text:' cell as representative of the labels. This is the longest name, and we want to make sure the shape is large enough to hold it.



You can bold, italicize, align to shape top, or align to shape bottom using the toggle buttons shown below. The default value of "centered" will automatically apply.



Color is an important aspect of any visualization, and the Style Designer provides complete support for all Graphviz color schemes.

Graphviz defines a color scheme as the context for interpreting color names.

"In particular, if a color value has form "xxx" or "//xxx", then the color xxx will be evaluated according to the current color scheme. If no color scheme is set, the standard X11 naming is used. For example, if colorscheme="bugn9", then color="7" is interpreted as "/bugn9/7".

Sounds complicated, doesn't it. Let's simplify it.

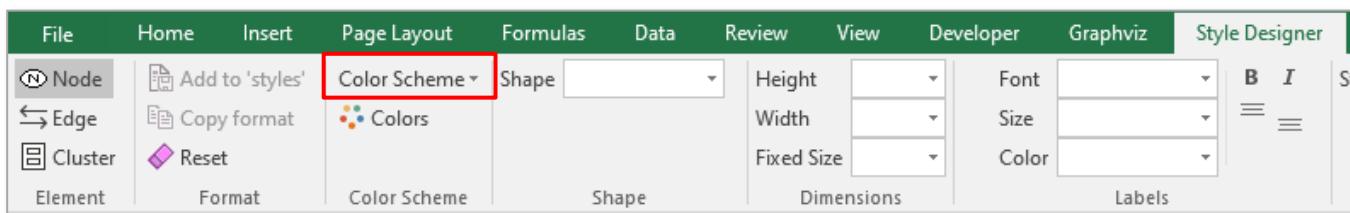
The 'Colors' button on the 'Style Designer' ribbon will show/hide the 'HELP - colors' worksheet. You will see on this worksheet that Graphviz supports 267 color schemes, and each color scheme has from 3 to 656 colors per scheme.

X11 Color Scheme		SVG Color Scheme		Brewer Color Schemes	
X11	Color Name	SVG	Color Name	Color Scheme	Color Name
	AliceBlue		AliceBlue	accent3	1 2 3
	AntiqueWhite		AntiqueWhite	accent4	1 2 3 4
	AntiqueWhite1		Aqua	accent5	1 2 3 4 5
	AntiqueWhite2		Aquamarine	accent6	1 2 3 4 5 6
	AntiqueWhite3		Azure	accent7	1 2 3 4 5 6 7
	AntiqueWhite4		Beige	accent8	1 2 3 4 5 6 7 8
	Aquamarine		Bisque		
	Aquamarine1		Black		
	Aquamarine2		BlanchedAlmond		
	Aquamarine3		Blue		
	Aquamarine4		BlueViolet		
	Azure		Brown		
	Azure1		Burlywood		
	Azure2		CadetBlue		
	Azure3		Chartreuse		
	Azure4		Chocolate		
	Beige		Coral		
	Bisque		CornflowerBlue		
	Bisque1		Cornsilk		
	Bisque2		Crimson		
	Bisque3		Cyan		
	Bisque4		DarkBlue		
	Black		DarkCyan		
	Blue		DarkGoldenrod		
	Blue1		DodgerBlue		

HELP - colors

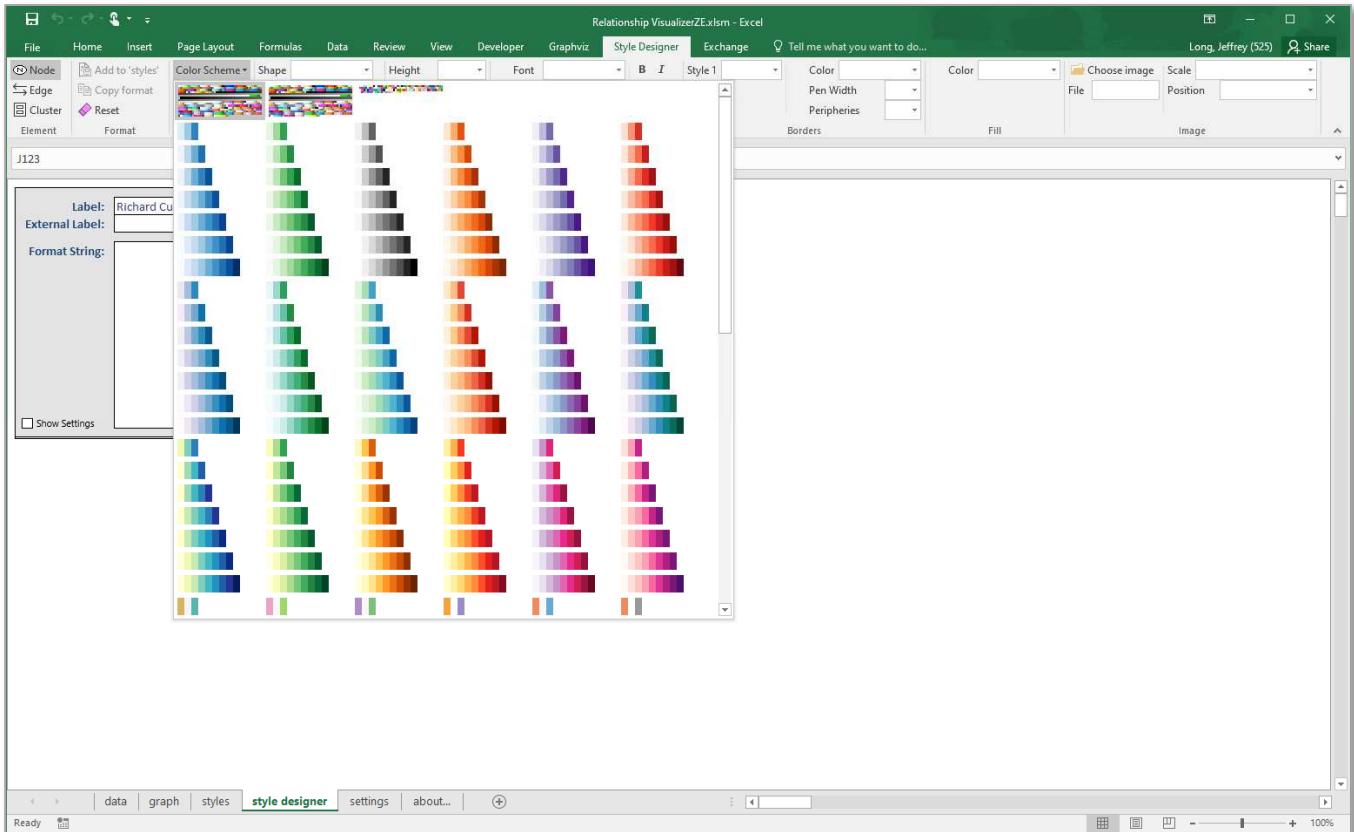
TIP: If you see a collection of colors you would like to use, simply click on the color scheme name hyperlink to transfer the color name to the 'style designer' worksheet ribbon.

This worksheet is used behind the scenes to create preview images for color choices. In the 'Style Designer' ribbon you will see a "Color Scheme" drop-down.



Excel to Graphviz for Win OS

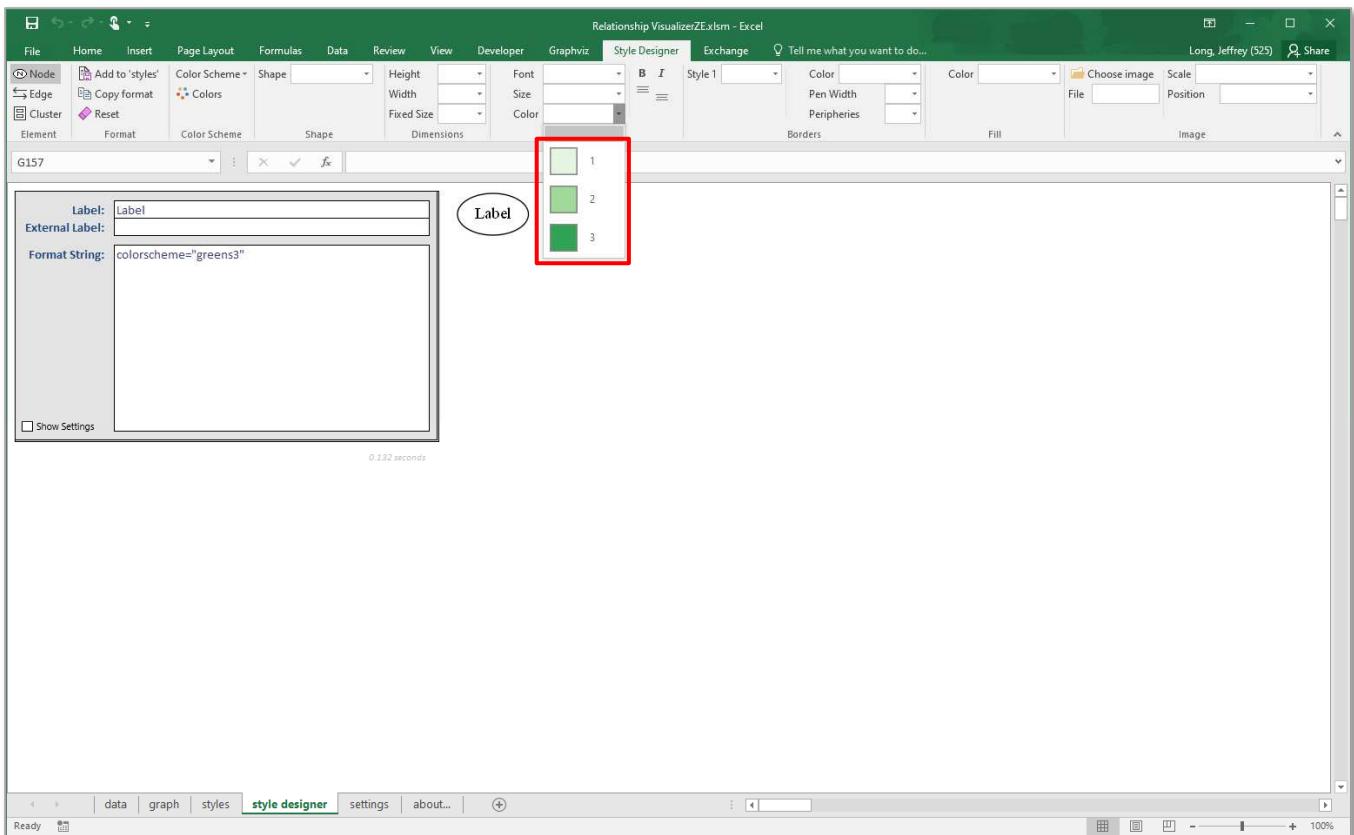
When you click on this drop down, a gallery where you can choose a color scheme appears such as:



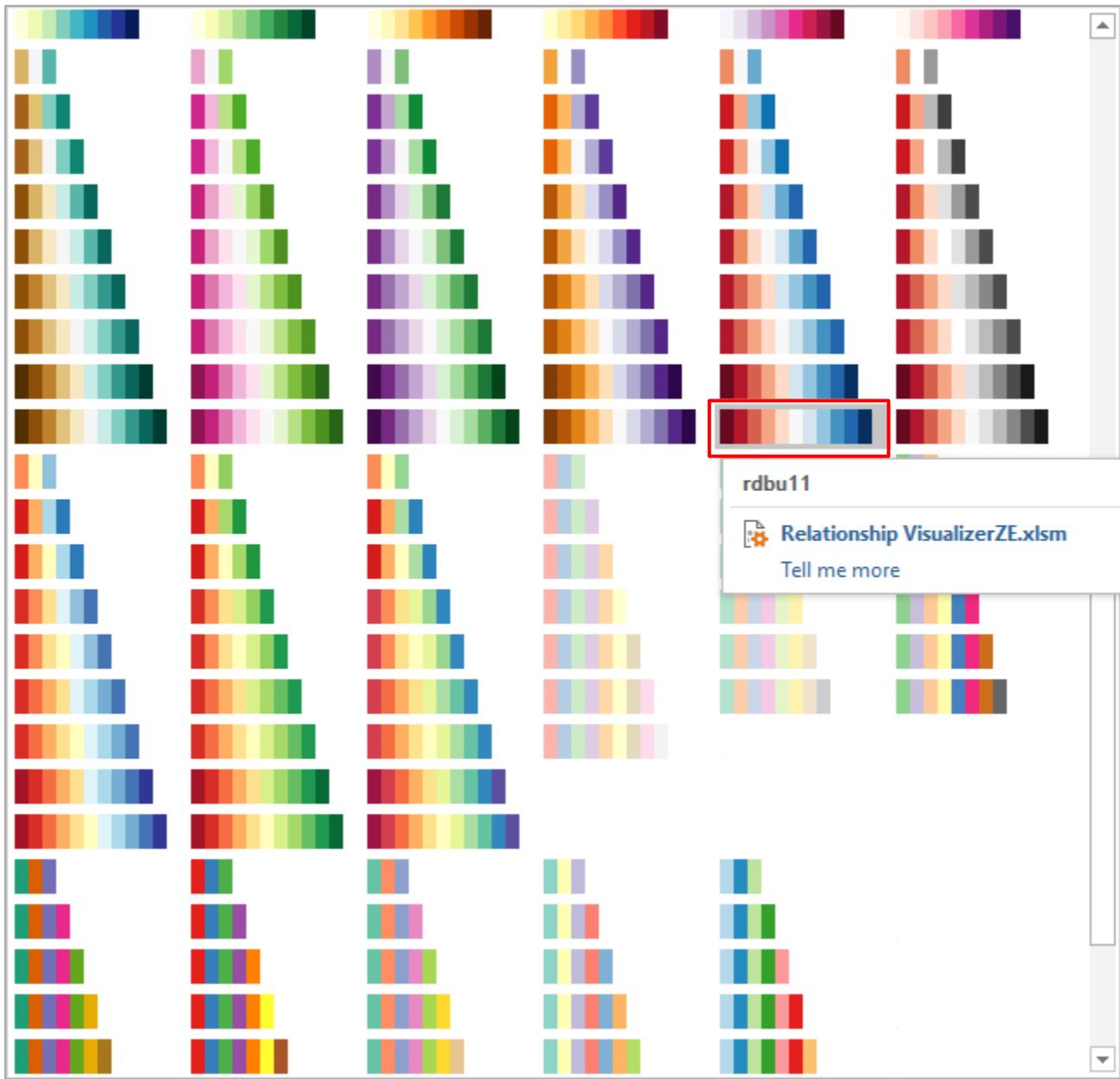
When you choose a color scheme, all the color related drop-down lists will be refreshed to contain the color names for that color scheme. For example, if you choose color scheme 'greens3' the dropdown lists will have the values '1', '2', and '3'.



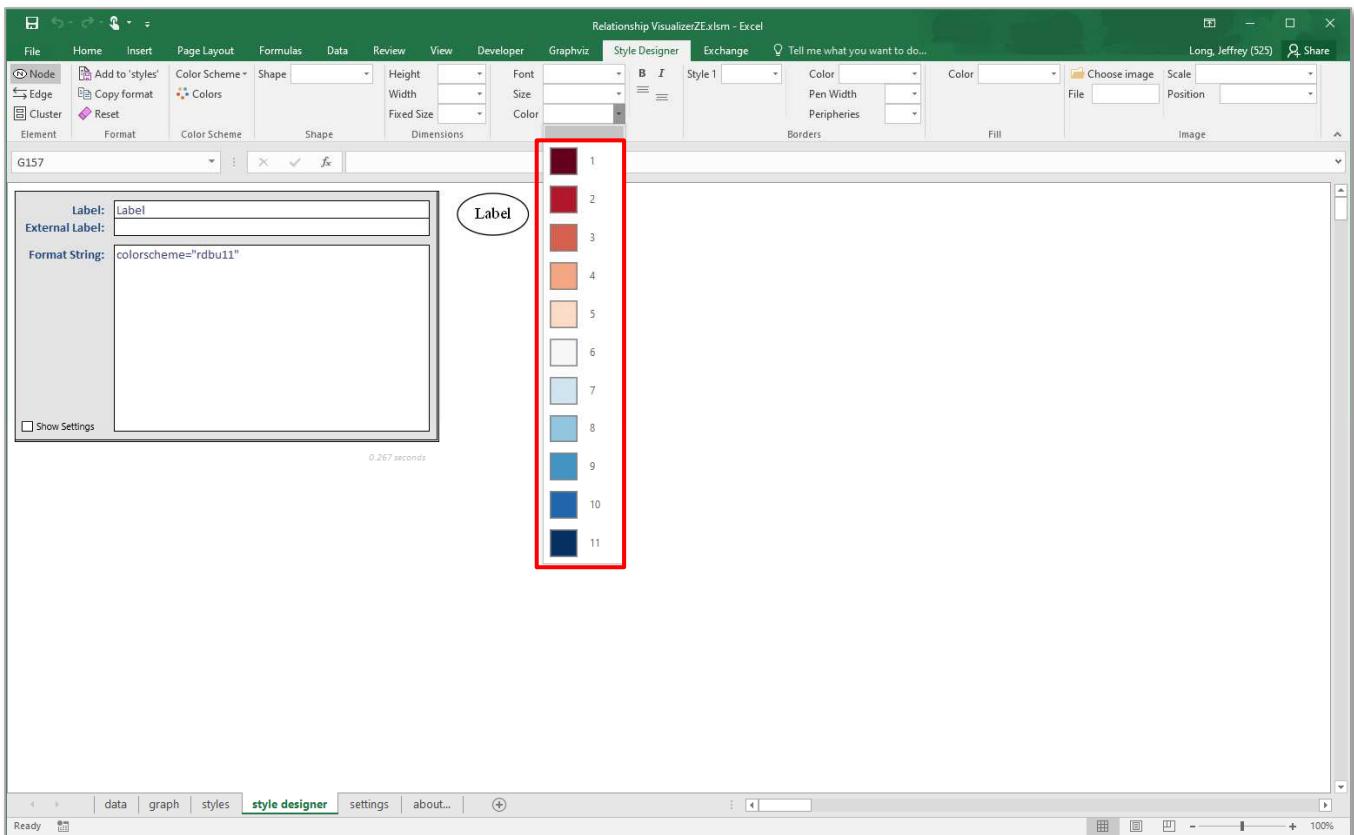
Excel to Graphviz for Win OS



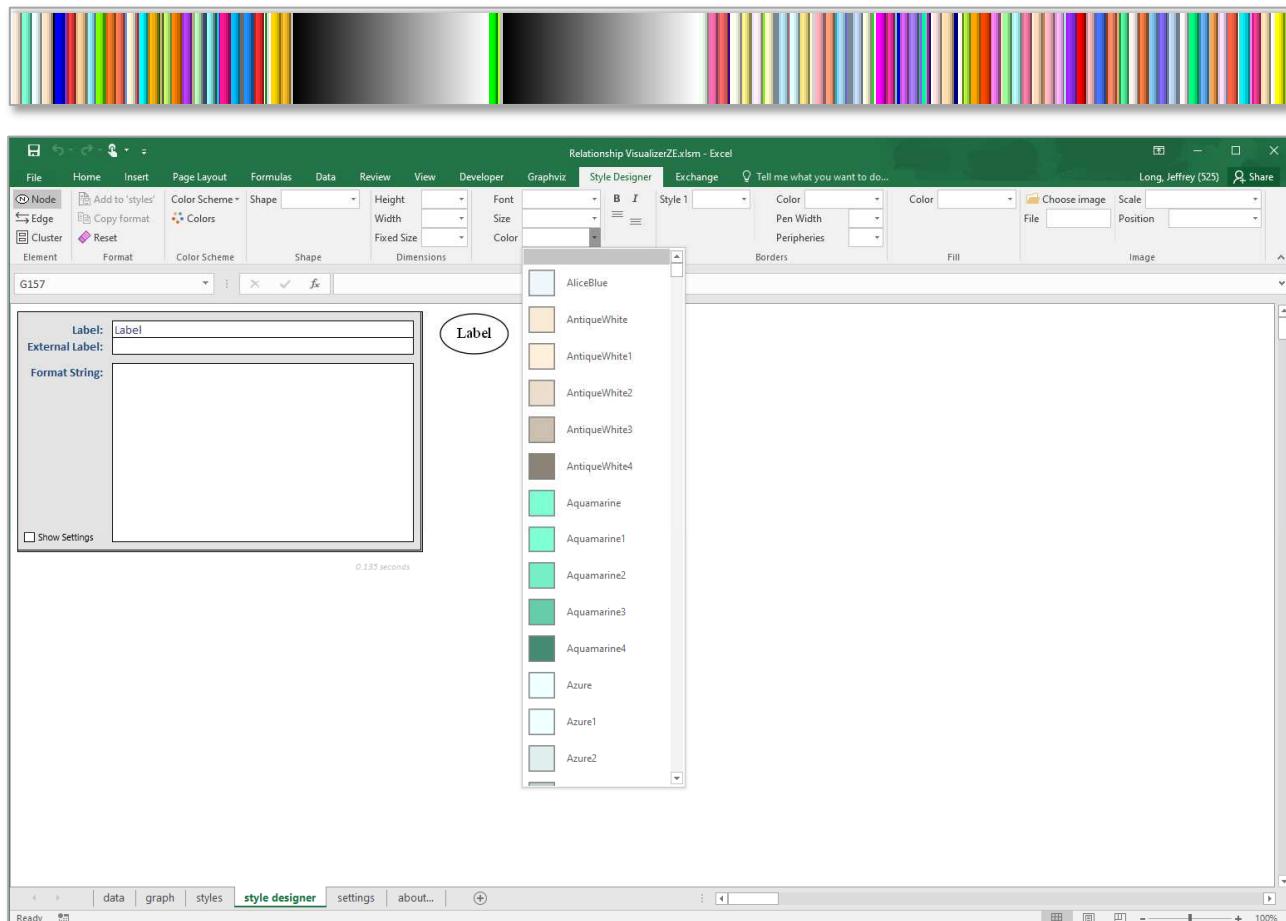
If you switch the color scheme, all the color related drop-down lists will be refreshed to contain the color names for the new color scheme. For example, if after choosing 'greens3' you choose color scheme 'rdbu11' the dropdown lists will have the values '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11'.



Excel to Graphviz for Win OS



Leave the "Color Scheme" setting as blank for now. When the Color Scheme setting is blank the default "X11" color scheme colors will be listed. X11 is the largest set of defined colors and provides 656 color choices to pick from.

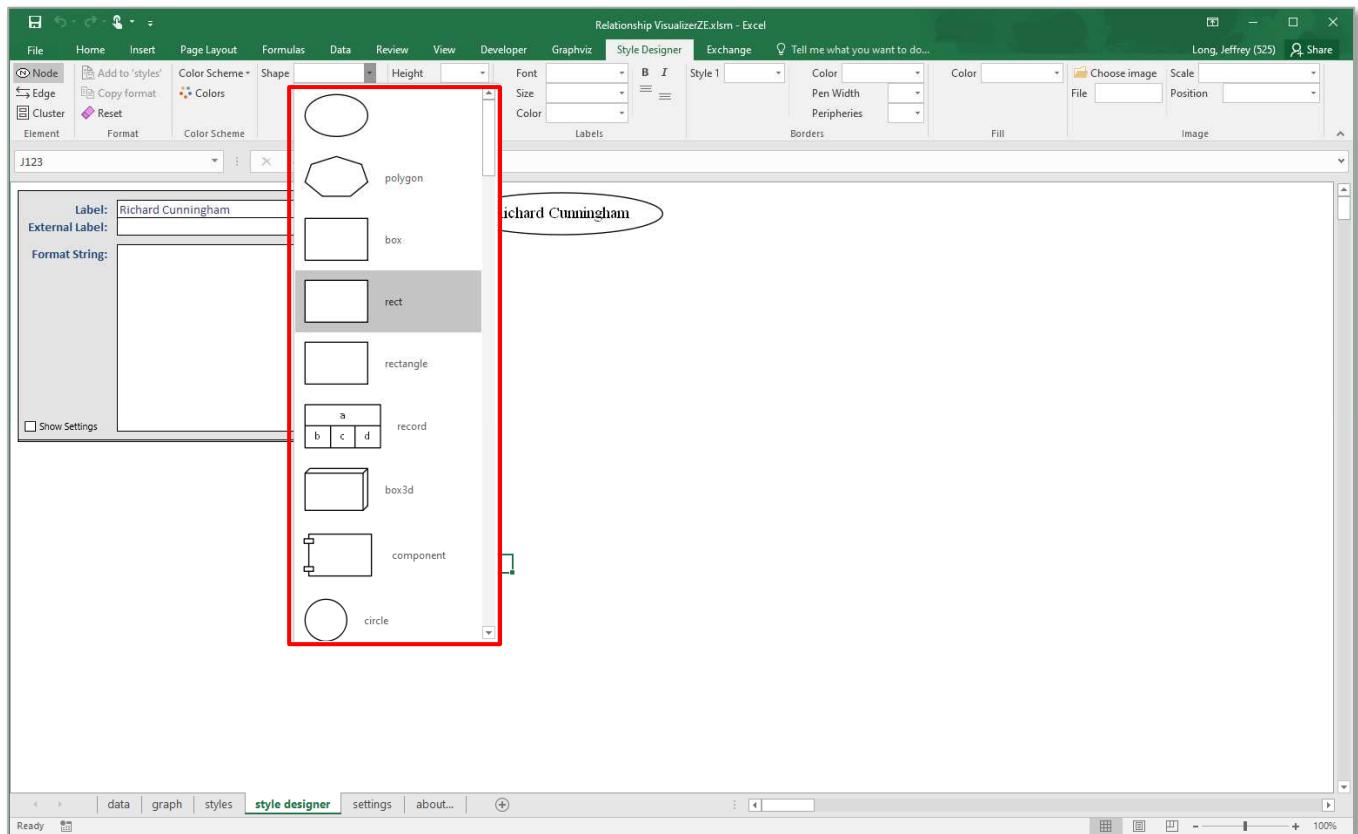


A drawback of having 656 color choices to choose from is that the X11 color scheme slows the loading of the ribbon. Dropdown list images such as the color scheme gallery images are stored in the spreadsheet, but Microsoft Excel requires using image files from the file system for dynamic lists such as the color names. A 20x20 pixel preview image for each color in a color scheme is created for the dropdown color lists the first time a color scheme is selected using color information in the "HELP - colors" worksheet cells. The status bar is updated as this process occurs to let you know that Excel is not frozen.

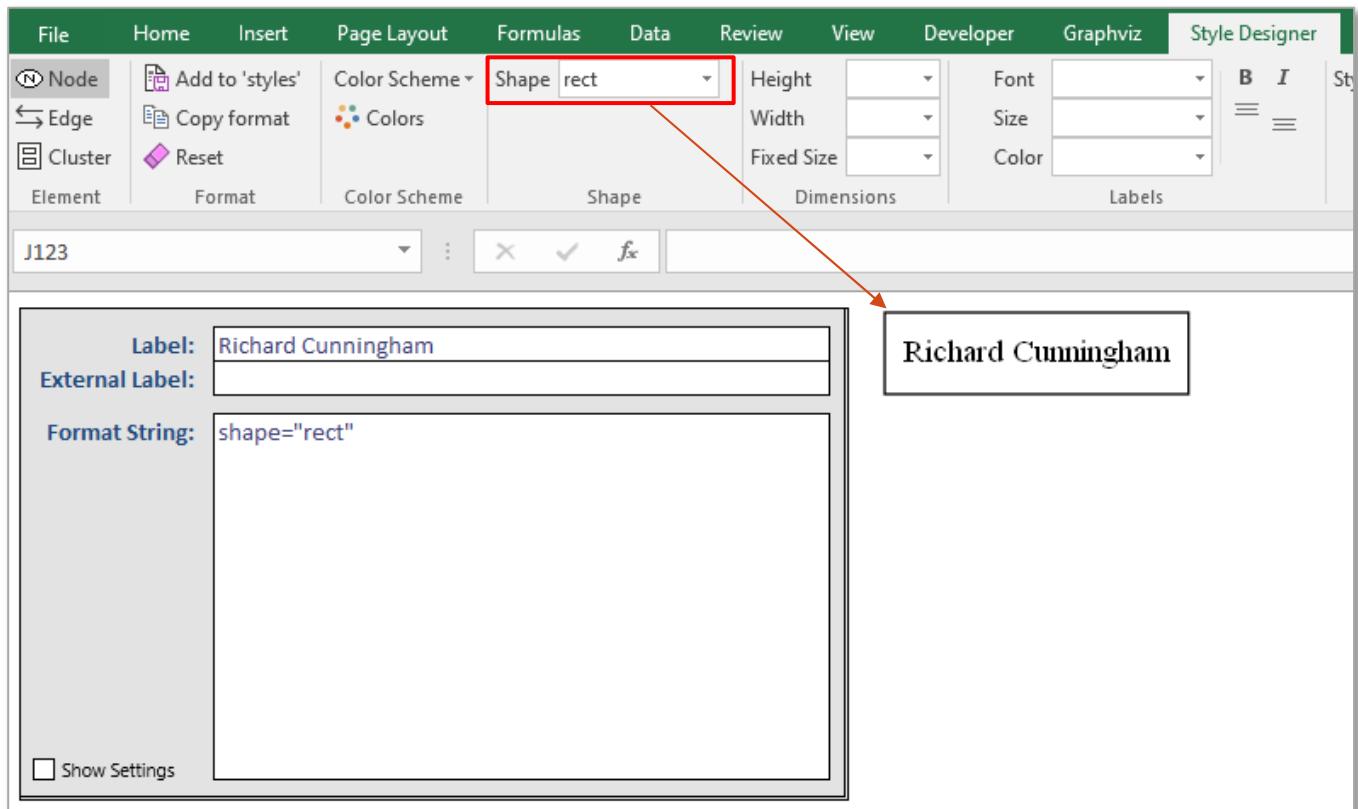
X11 is Graphviz's default color scheme, resulting in 656 preview images being created one time. The names of all 656 colors also have to be added to four dropdown lists of color names (border color, font color, fill color, and gradient fill color, i.e., 2,624 list entries). The time to display the ribbon can range from 2-30 seconds depending upon the speed and power of your computer.

Excel to Graphviz for Win OS

Click on the 'Node Shape' drop-down list. A list of shapes supported by Graphviz is presented along with a sample image of the shape. Specify the 'Node Shape' as "rect" (i.e., rectangle).



Note that the format string now says shape="rect" and the node shape changes to a rectangle in the preview.



Excel to Graphviz for Win OS

Up to 3 border styles are selectable and are additive making it possible to have styles such as bold edge and rounded corners. When you click on any of the 'Border Style' drop-down lists you will be presented with the list of choices along with a sample image of the style.

Set the 'Border Style 1' as "rounded" to give the rectangle rounded corners.

The screenshot shows the Microsoft Graphviz ribbon interface. The 'Style Designer' tab is selected. In the 'Border Style 1' dropdown, the 'rounded' option is highlighted. In the 'Style 1' dropdown under the 'Style Designer' tab, 'rounded' is also selected. A red arrow points from the 'Format String' field in the bottom-left to a rounded rectangle node labeled 'Richard Cunningham' on the canvas.

File Home Insert Page Layout Formulas Data Review View Developer Graphviz Style Designer Exchange ? Tell r

Node Add to 'styles'
Edge Copy format Cluster Reset Element Format Color Scheme Colors

Shape rect Height Width Fixed Size

Font Size Color

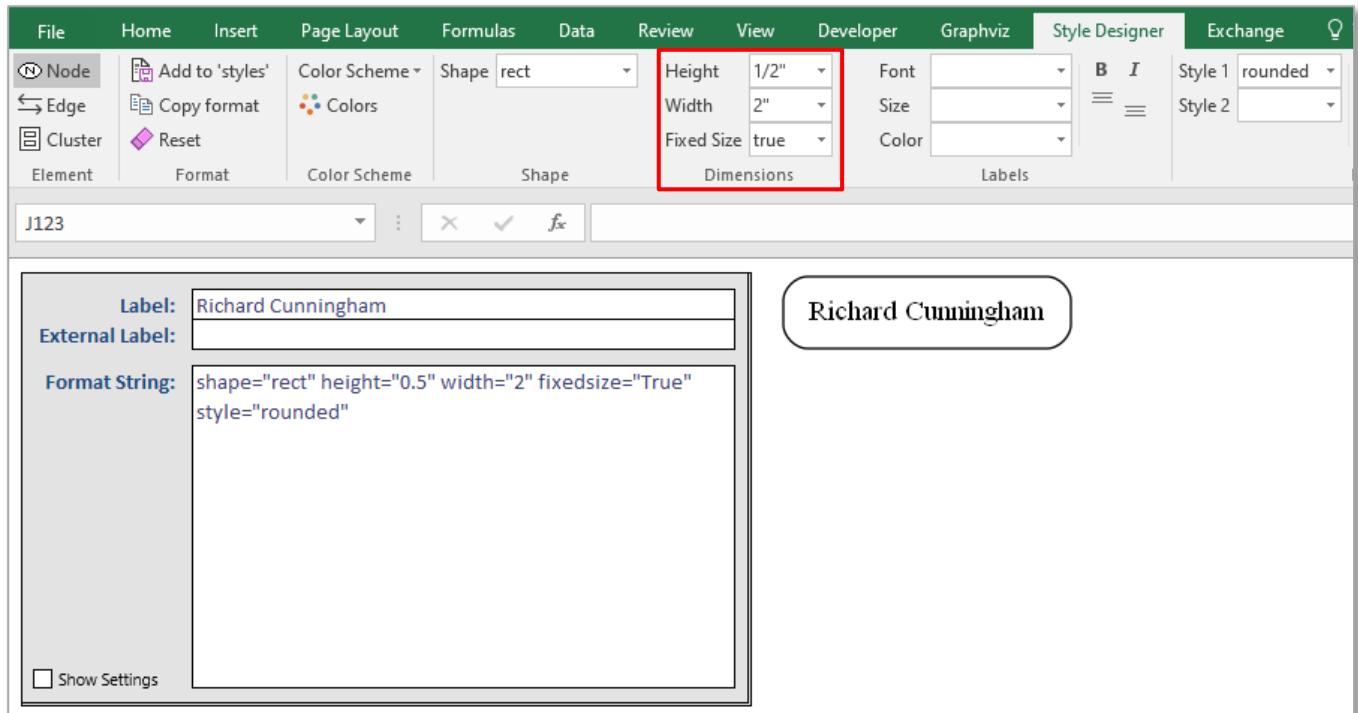
B I Style 1 rounded
Style 2

J123

Label: Richard Cunningham
External Label:
Format String: shape="rect" style="rounded"
 Show Settings

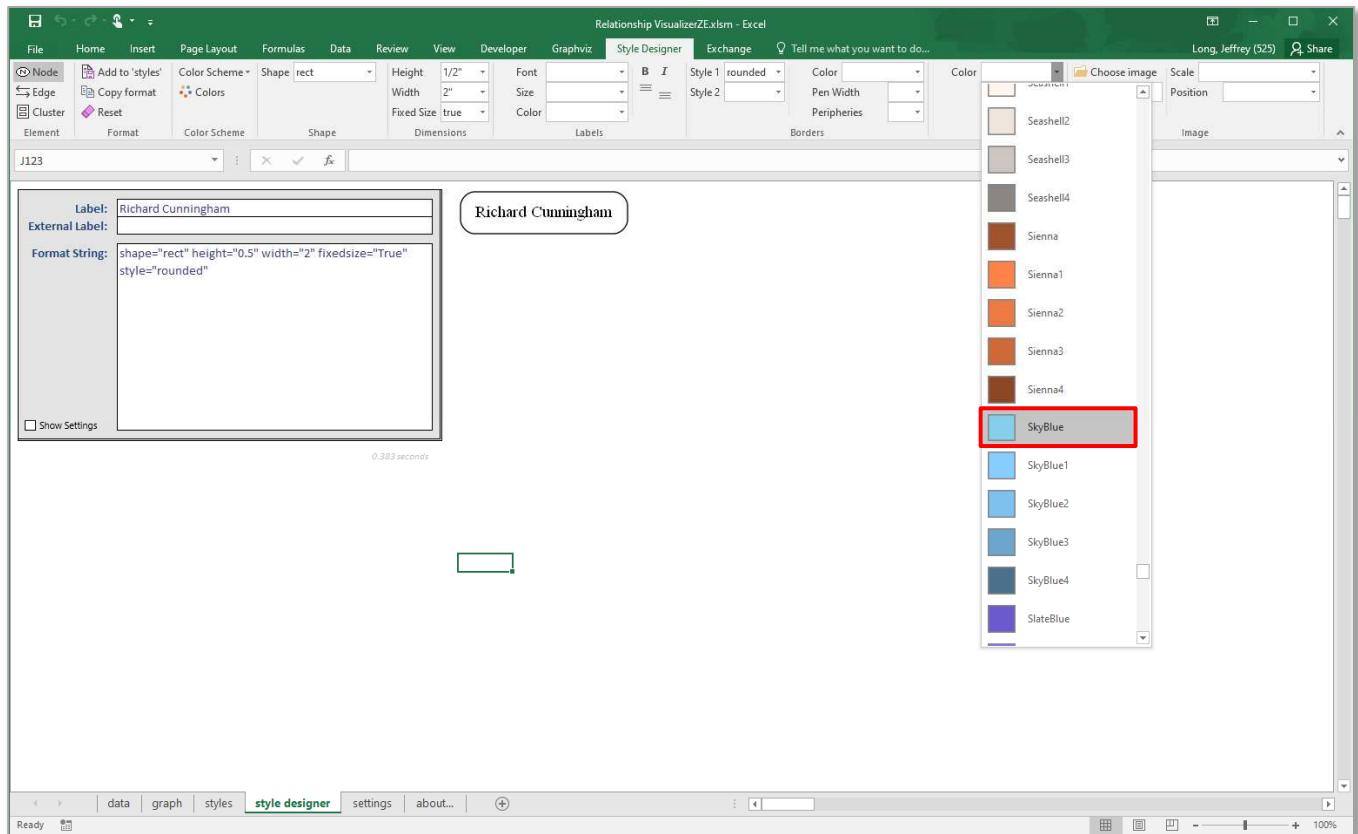
Richard Cunningham

Set the 'Height:' to $\frac{1}{2}$ " and 'Width:' to 2" inches (inches are Graphviz's unit of measure). Also, set 'Fixed Size' to "true" to lock the size of the shape. Leave 'Fixed Size' blank or set it to "false" to allow the shape to grow if the label size exceeds the height and width specified.

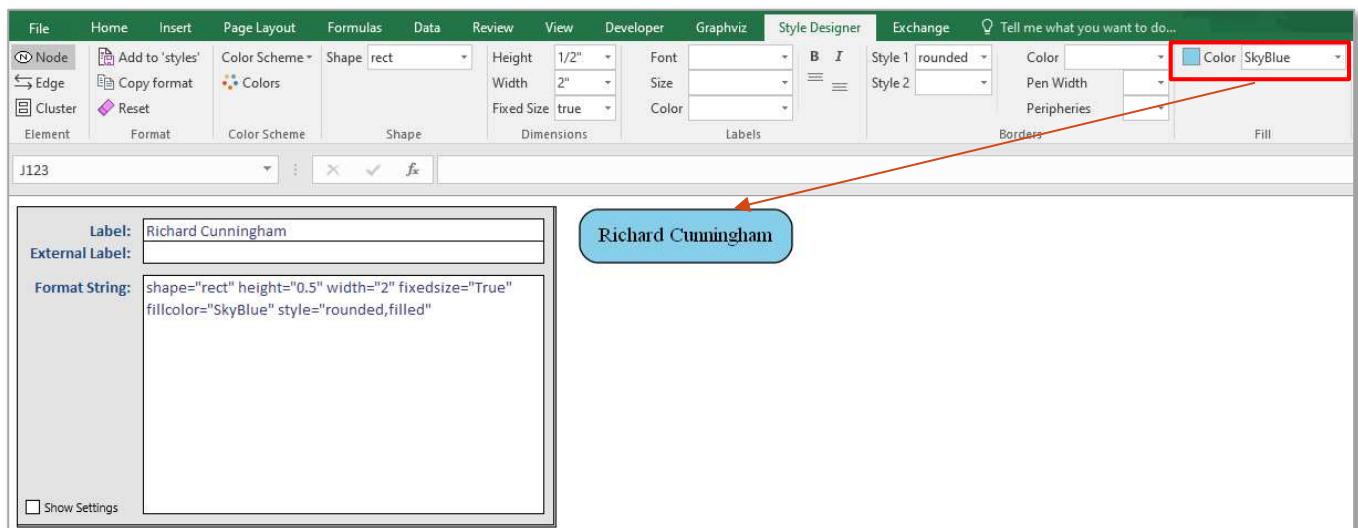


Excel to Graphviz for Win OS

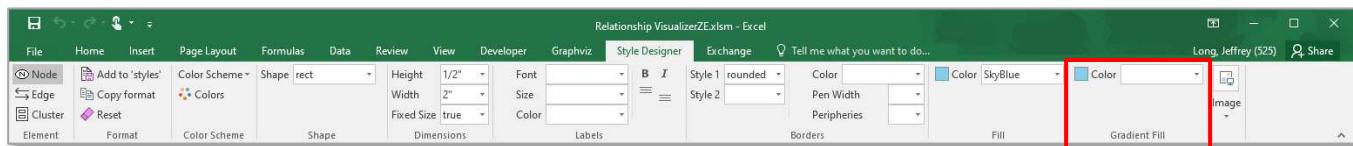
Set the 'Fill Color:' to "SkyBlue".



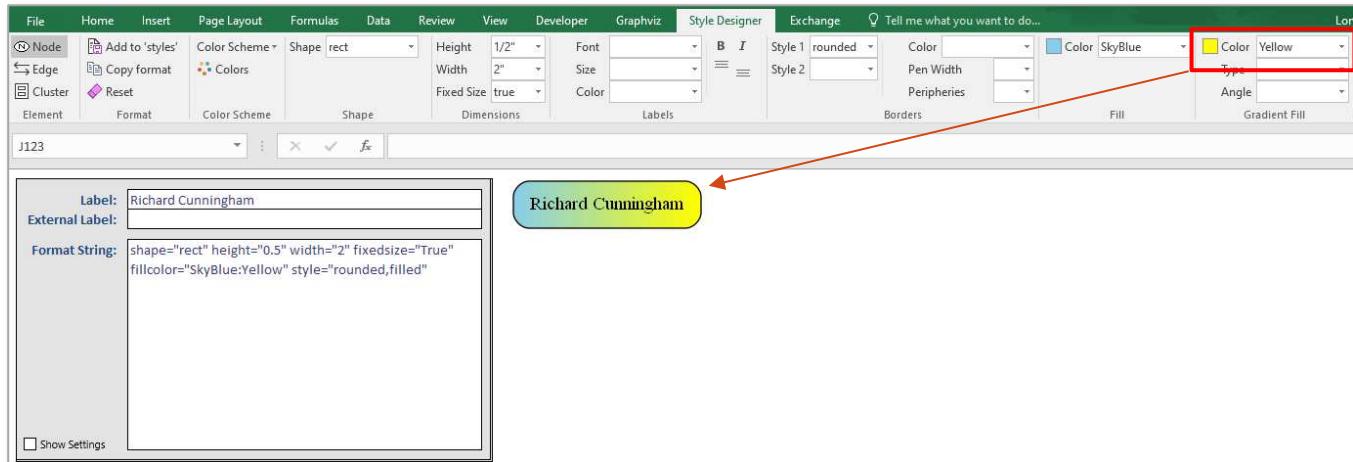
This setting will give the shape a background color, and automatically add "filled" to the list of 'Style' attributes. The preview image changes to look like:



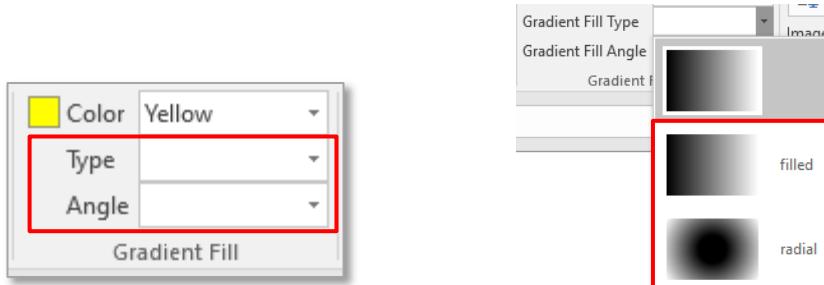
Notice that the ribbon dynamically changes once a Fill Color is specified to display a new choice for 'Gradient Fill Color'.



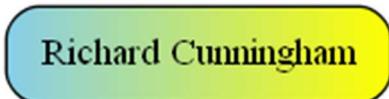
A "Gradient Fill Color" allows you to select a second color which the Fill Color will gradually transition to. If you select "yellow" as the Gradient Fill Color the preview image changes to look like:



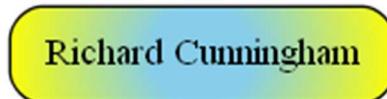
Another set of dynamic changes occur as two additional choices 'Type' and 'Angle' appear beneath Gradient Fill Color which allow you to define how the transition occurs. The Gradient Type is either "filled" (i.e., linear) or "radial".



The differences are illustrated below:



Gradient Type: filled



Gradient Type: radial

Changing the Gradient Angle moves the angle of the gradient fill. For linear fills, the colors transform along a line specified by the angle and the center of the object. For radial fills, a value of zero causes the colors to transform radially from the center; for non-zero values, the colors transform from a point near the object's periphery as specified by the value.

If you change the Gradient Angle to 180 degrees, the preview images now appear as:

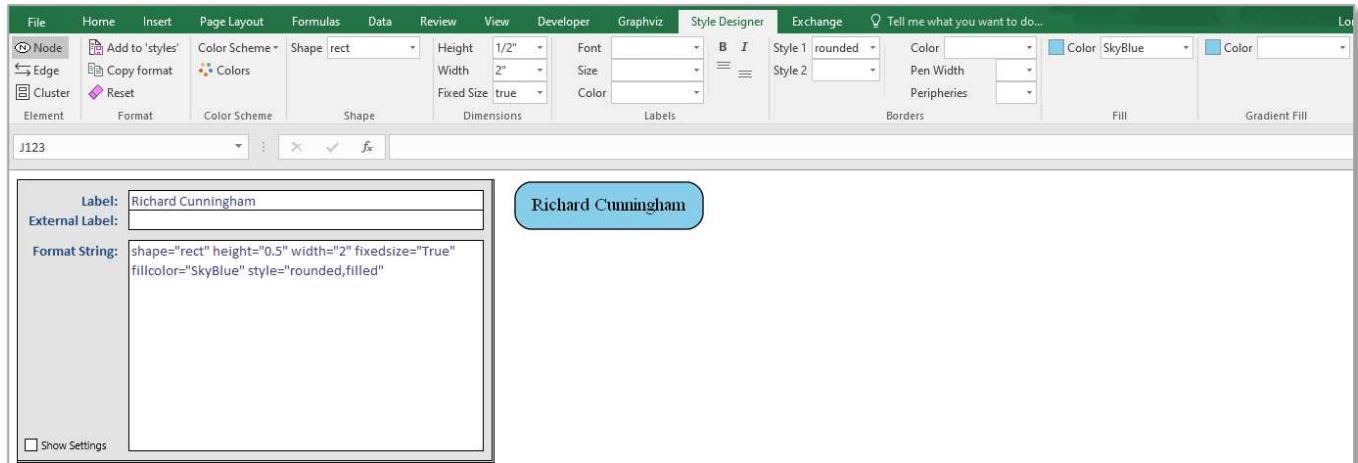
Richard Cunningham

Richard Cunningham

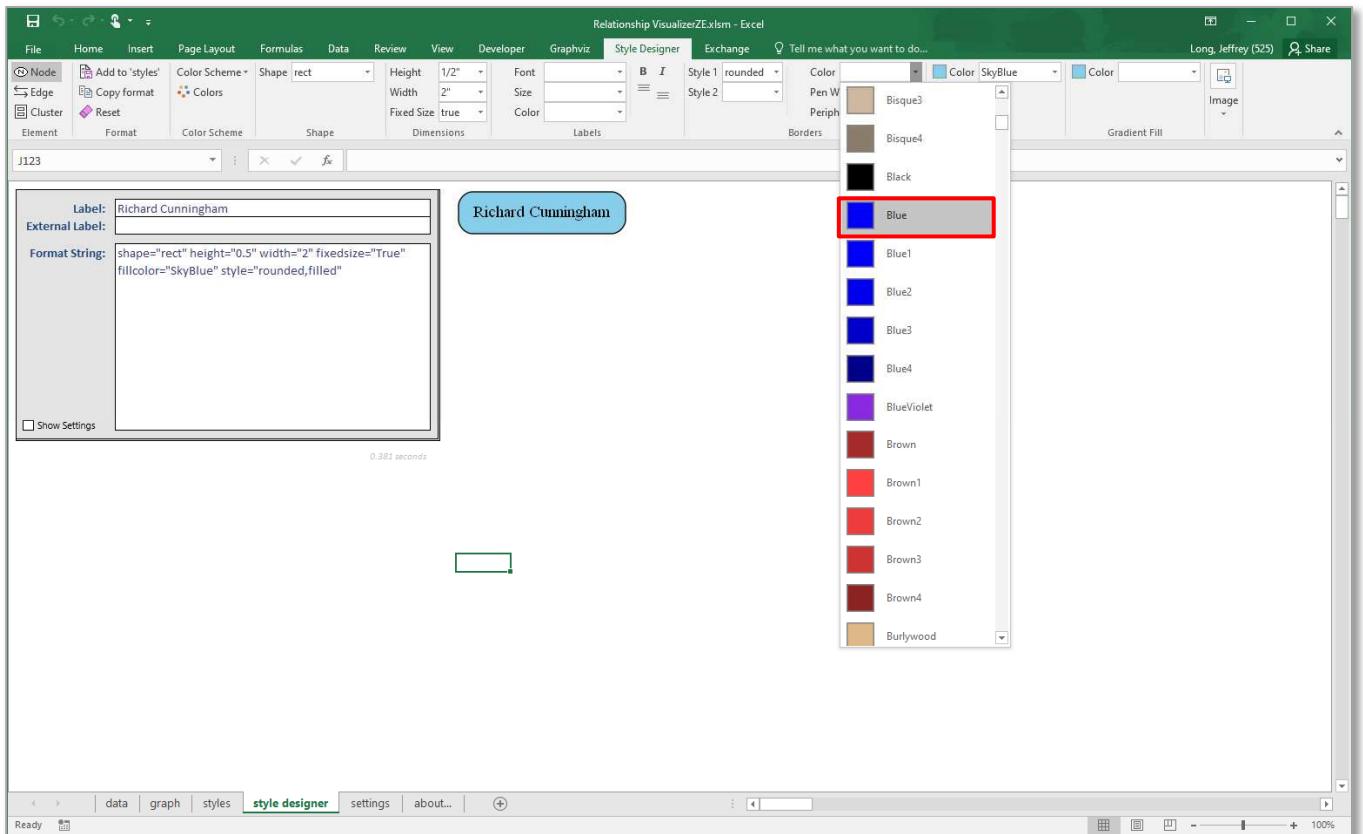
Gradient Type: filled, Gradient Angle: 180

Gradient Type: radial, Gradient Angle: 180

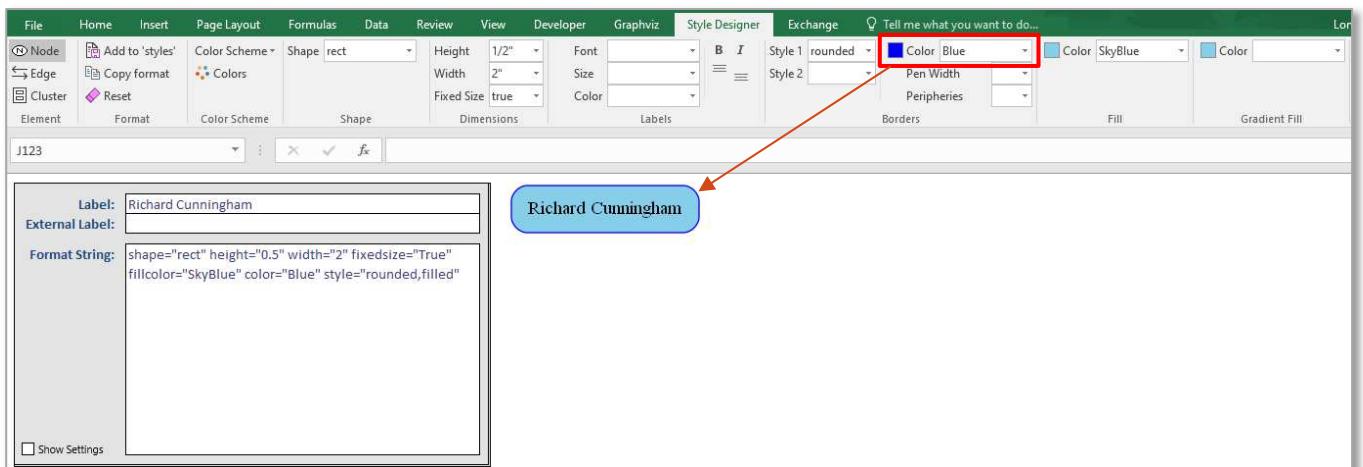
For our Happy Days example, we do not want to use Gradient Fills as they will not add much visual impact to this example. Delete "yellow" from the 'Gradient Fill Color:' choice. Notice that the preview image reverts to the solid 'skyblue' background, and the options for Gradient Type, and Gradient Angle have disappeared. The preview image now looks as follows:



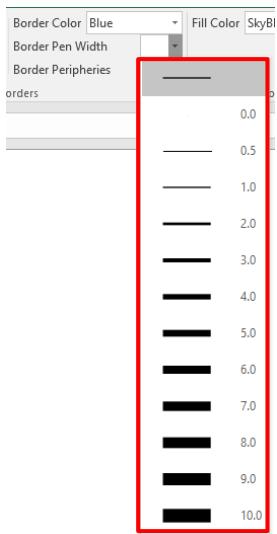
Set the 'Border Color' to 'blue'.



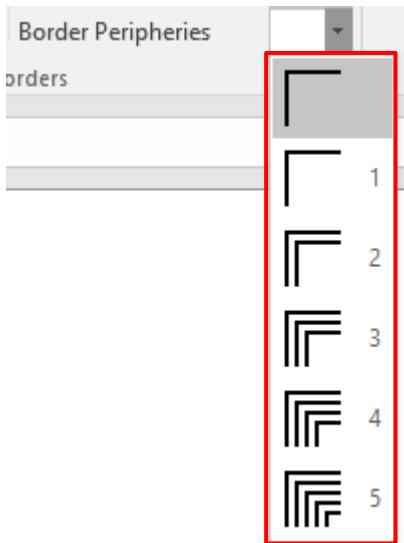
The preview image now appears as follows:



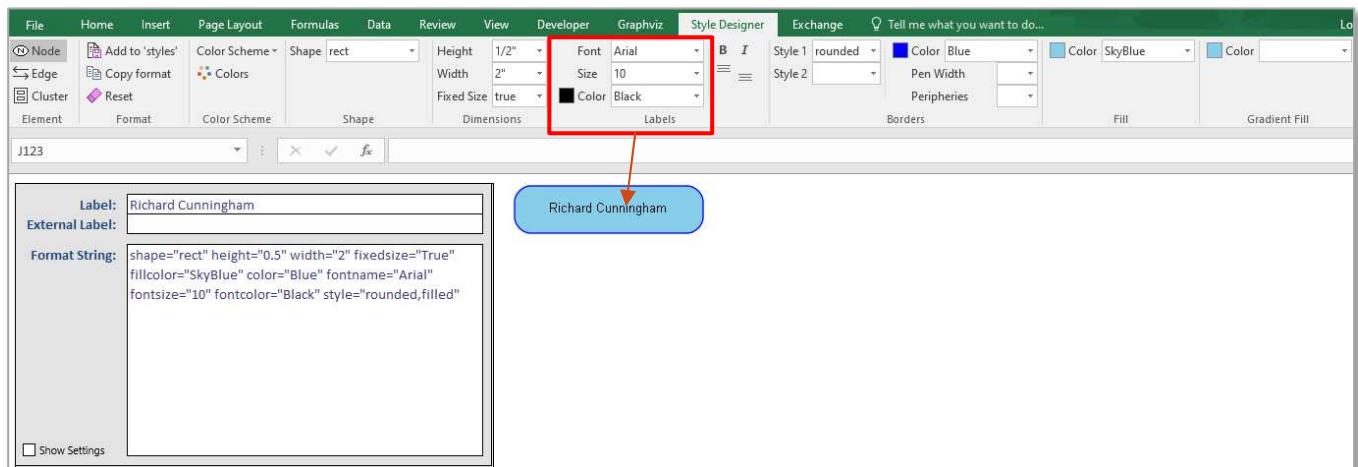
Border Pen Width can make the line around the shape thicker. We are content with the current thickness, so leave the 'Border Pen Width' setting blank to use the default border line width.



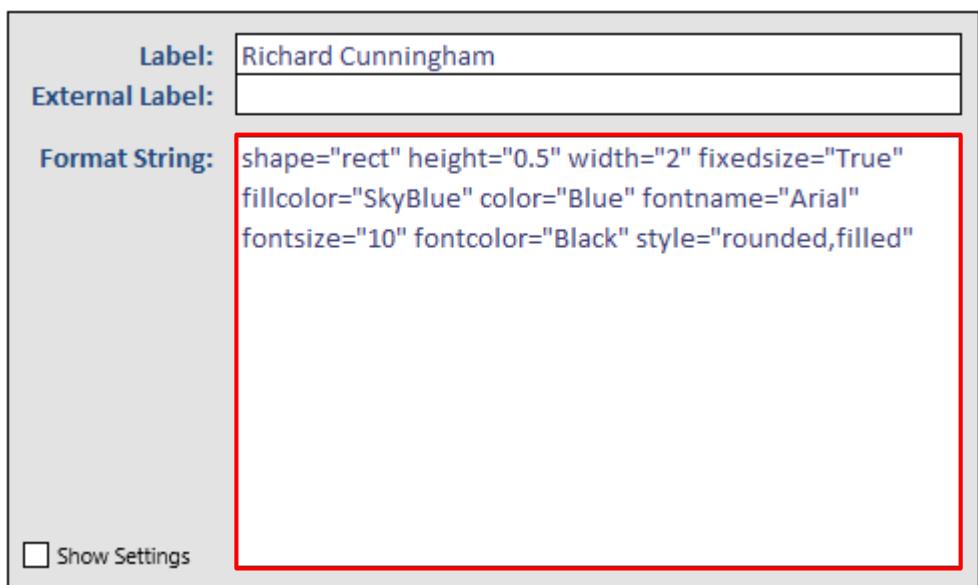
'Border Peripheries' can be used to add concentric borders to the shape. Leave the 'Border Peripheries' setting blank to maintain a single border around the shape.



For the label, set the 'Font Name:' to "Arial", the 'Font Size:' to "10" points, and the 'Font Color:' to "black". The preview image now appears as follows:



Notice that as we have been making selections the format string cell has been building a list of Graphviz style attributes in the cell highlighted below:



This cell is also an active cell. You can make edits directly into this cell to tweak settings to use values not supplied in the dropdown lists. For example, the font size list jumps from 36 to 48, but if you want a font size of 40 you may type in the value.

Be aware that any change in the ribbon will cause any hand-made edits in the format string to be wiped out as ribbon changes result in a new format string being built. Conversely, deleting the contents of the format string cell will result in the ribbon settings getting reset to default values.

Using the 'styles' Worksheet

The "styles" worksheet is where you can create style definitions for nodes and edges. It works in theory in a manner like an HTML Cascading Style Sheet where you can define a style name, and how the style should appear (shape, color, font, etc.). A defined style can then be associated with many nodes or edges in the "data" worksheet.

The default "styles" worksheet appears as follows:

A	B	C	D	E	F
	Style Name	Format	Style Type	All Styles	No Clusters
1					
2	node		node	yes	yes
3	edge		edge	yes	yes
4	subgraph-open		subgraph-open	yes	no
5	subgraph-close		subgraph-close	yes	no
6	keyword		keyword	yes	yes
7	native		native	yes	yes
8	Border 0 Begin	penwidth="2" fontname="Calibri Bold" fontsize="16" margin="18"	subgraph-open	yes	no
9	Border 0 End		subgraph-close	yes	no
10	Border 1 Begin	penwidth="1" colorscheme="pastel16" fillcolor="6" fontname="Calibri Bold" fontsize="12" style="filled" margin="18"	subgraph-open	yes	no
11	Border 1 End		subgraph-close	yes	no
12	Border 2 Begin	penwidth="1" colorscheme="pastel16" fillcolor="5" fontname="Calibri Bold" fontsize="12" style="filled" margin="18"	subgraph-open	yes	no
13	Border 2 End		subgraph-close	yes	no
14	Border 3 Begin	penwidth="1" colorscheme="pastel16" fillcolor="4" fontname="Calibri Bold" fontsize="12" style="filled" margin="18"	subgraph-open	yes	no
15	Border 3 End		subgraph-close	yes	no
16	Border 4 Begin	penwidth="1" colorscheme="pastel16" fillcolor="3" fontname="Calibri Bold" fontsize="12" style="filled" margin="18"	subgraph-open	yes	no
17	Border 4 End		subgraph-close	yes	no
18	Border 5 Begin	penwidth="1" colorscheme="pastel16" fillcolor="2" fontname="Calibri Bold" fontsize="12" style="filled" margin="18"	subgraph-open	yes	no
19	Border 5 End		subgraph-close	yes	no
20	Component	shape="component" height="0.75" width="1.25" fixedsize="True" fillcolor="White" fontname="Calibri" fontsize="10" style="filled"	node	yes	yes

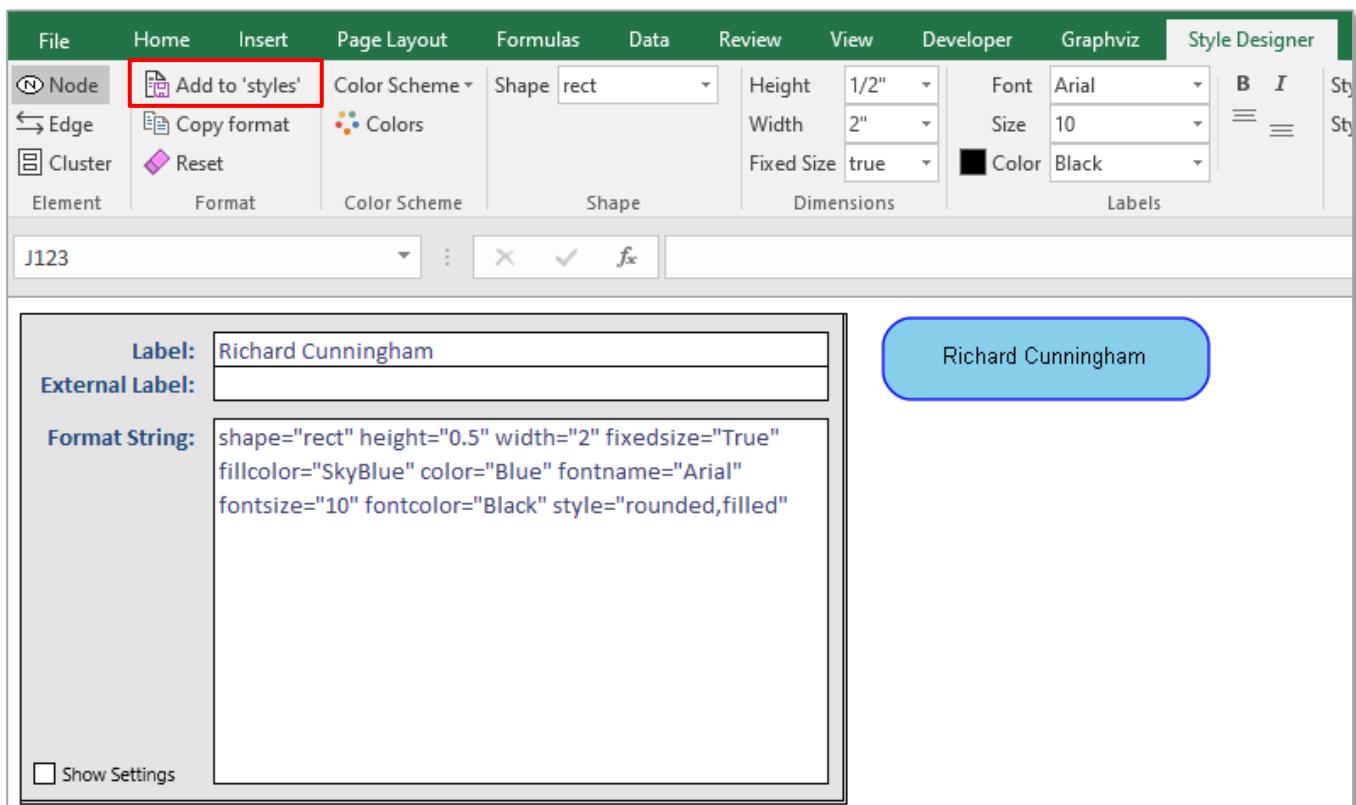
The columns are as follows:

- Column A - The "**Indicator**" column. It allows you to place a "#" character to denote a comment. It can be used to comment out the style, so it is excluded from the renderings.
- Column B - The "**Style**" column. This column is where you specify the Style name.
- Column C - The "**Format**" column. This column is where you paste the style definitions created on the 'style designer' worksheet to create the visual definition for any graph elements which get associated with this style in the 'data' worksheet.
- Column D - The "**Style Type**" column. This column must have a value of "node", "edge", "subgraph-open", "subgraph-close", "keyword", or "native". This string value tells the macros inside the Relationship Visualizer how to interpret the row and convert it into DOT language commands.
- Column E and beyond - The view switch columns. These columns are used for creating views of the data. It must contain a "Yes" or "No" value to indicate if the style should be included in the graph. This column's use is discussed in **Creating Views** beginning on page 145. All spreadsheets created from the Relationship Visualizer Excel template will have the Column E heading row set to "All", with "yes" values set for the style switches. This means that all styles should be included in the graphs when Column E controls the view.

You may have noticed that the Style names in column B have a dark border around their outer edge. This is a visual cue of an Excel named-range used in the 'data' worksheet which allows you select from the styles in the "Style" column.

Excel to Graphviz for Win OS

Now that you understand the purpose of the "styles" worksheet, let's add in the style definition we just created for the Cunningham family. Select the Add to 'styles' button to transfer the style to the 'styles' worksheet.



- Step 1** - A row has been inserted at the bottom of the list within the range marked by the dark border so that the style name is contained within the named range and will appear in the dropdown list of style choices on the 'data' worksheet. The "Style Name" cell is selected, and you may type over the temporary name (e.g., Node style 21) which has been assigned.

	A	B	C	D	E	F
	Style Name	Format	Style Type	All Styles	No Clusters	
1						
38	Point	shape="point"	node	yes	yes	
39	Predefined Process	shape="rectangle" height="0.75" width="1.25" fixedsize="True" fillcolor="White" fontname="Calibri" fontsize="10" style="diagonals,filled"	node	yes	yes	
40	Preparation	shape="hexagon" height="0.75" width="1.25" fixedsize="True" fillcolor="White" fontname="Calibri" fontsize="10" style="filled"	node	yes	yes	
41	Process	shape="rect" height="0.75" width="1.25" fixedsize="True" fillcolor="White" fontname="Calibri" fontsize="10" style="filled"	node	yes	yes	
42	Small Circle	shape="circle" height="0.25" fixedsize="True" style="filled" penwidth="1" fontname="Arial" fontsize="8"	node	yes	yes	
43	Stop	shape="octagon" height="0.75" width="0.75" fixedsize="True" color="Red" fillcolor="white" fontname="Calibri" fontsize="10" style="filled"	node	yes	yes	
44	Node style 21	shape="rect" height="0.5" width="2" fixedsize="True"	node	yes	yes	
45						

- Step 2** - Type "Cunningham Family" as the "Style Name" name
- Step 3** - Observe that the style definition has been copied from the 'Style Designer'.
- Step 4** - Observe that the "Style Type" value has been set to "node"
- Step 5** - Observe that the switches in the remaining columns are all set to "yes".

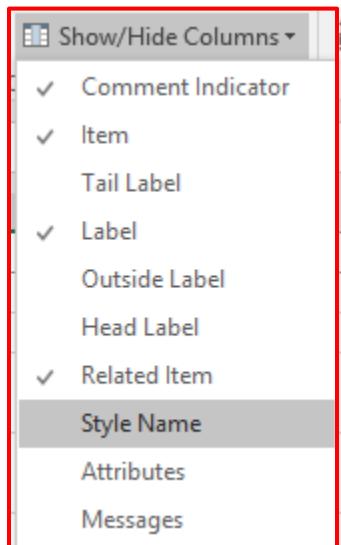
The 'styles' worksheet should now look as follows:

	A	B	C	D	E	F
1		Style Name	Format	Style Type	All styles	No Clusters
38	Point	shape="point"		node	yes	yes
39	Predefined Process	shape="rectangle" height="0.75" width="1.25" fixedsize="True" fillcolor="White" fontname="Calibri" fontsize="10" style="diagonals,filled"		node	yes	yes
40	Preparation	shape="hexagon" height="0.75" width="1.25" fixedsize="True" fillcolor="White" fontname="Calibri" fontsize="10" style="filled"		node	yes	yes
41	Process	shape="rect" height="0.75" width="1.25" fixedsize="True" fillcolor="White" fontname="Calibri" fontsize="10" style="filled"		node	yes	yes
42	Small Circle	shape="circle" height="0.25" fixedsize="True" style="filled" penwidth="1" fontname="Arial" fontsize="8"		node	yes	yes
43	Stop	shape="octagon" height="0.75" width="0.75" fixedsize="True" color="Red" fillcolor="white" fontname="Calibri" fontsize="10" style="filled"		node	yes	yes
44	Cunningham Family	shape="rect" height="0.5" width="2" fixedsize="True" fillcolor="SkyBlue" color="Blue" fontname="Arial" fontsize="10" fontcolor="Black" style="rounded,filled"		node	yes	yes

Next, we will return to the "data" worksheet and use the new style definition.

Applying Custom Styles

Return to the "data" worksheet and unhide the 'Style Name' column by checking 'Style Name' on the 'Show/Hide Columns' dropdown list on the Graphviz tab.



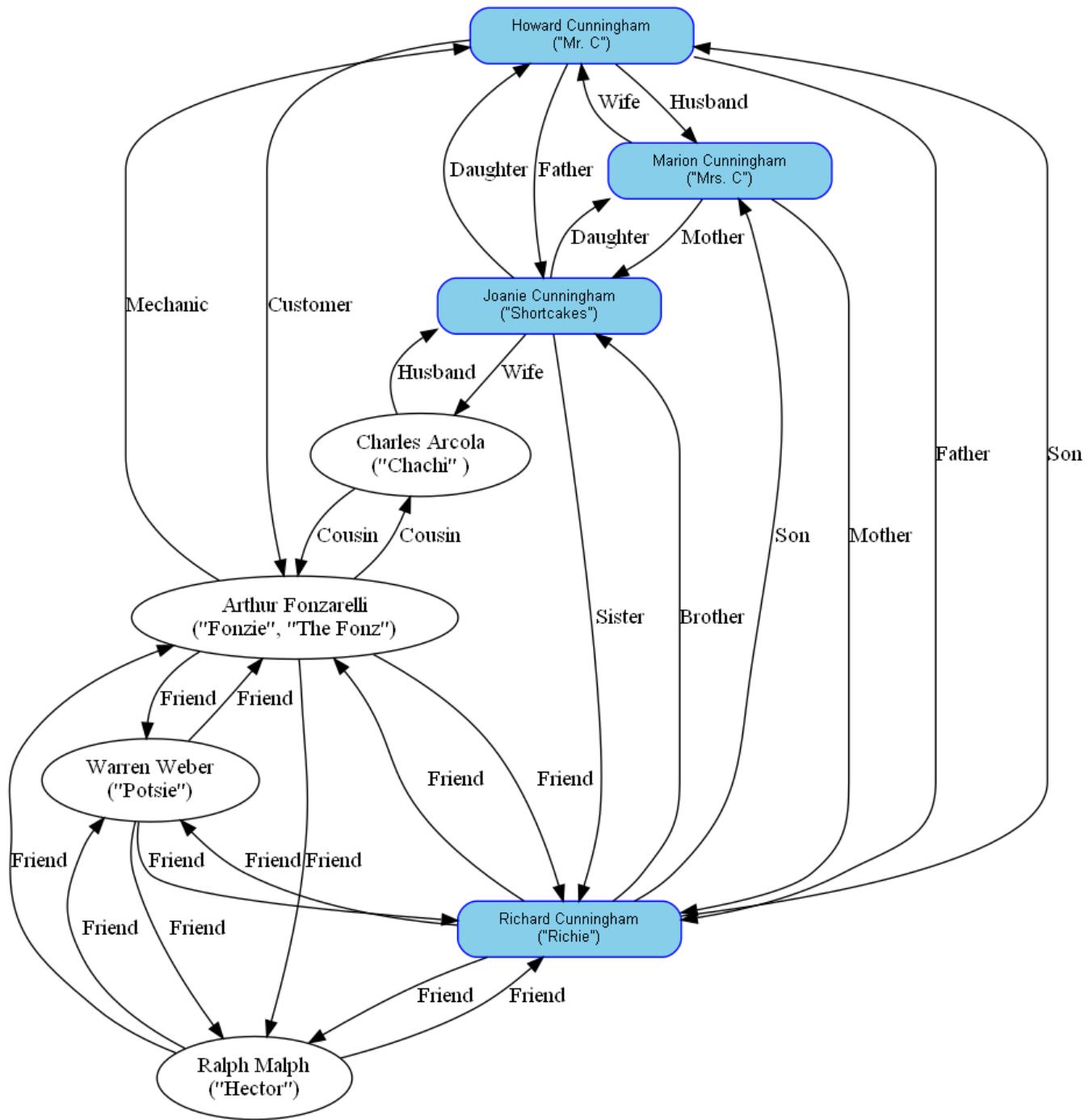
If you click on a cell in the "Style" column you should see that "Cunningham" family is now a choice in the dropdown list:

	A	B	D	G	H
1	Item	Label	Related Item	Style Name	
2					
3					
4	Howard	Howard Cunningham ("Mr. C")			Process
5	Marion	Marion Cunningham ("Mrs. C")			Small Circle
6	Richie	Richard Cunningham ("Richie")			Stop
7	Joanie	Joanie Cunningham ("Shortcakes")			Cunningham Family
8	Fonzie	Arthur Fonzarelli ("Fonzie", "The Fonz")			
9	Chachi	Charles Arcola ("Chachi")			
10	Ralph	Ralph Malph ("Hector")			
11	Potsie	Warren Weber ("Potsie")			

Change the style from "node" to "Cunningham Family" for Howard, Marion, Richard, and Joanie Cunningham. The "data" worksheet should now look as follows:

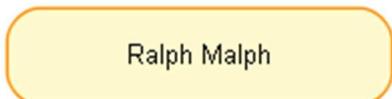
A	B	D	G	H
1	Item	Label	Related Item	Style Name
2				
3				
4	Howard	Howard Cunningham ("Mr. C")		Cunningham Family
5	Marion	Marion Cunningham ("Mrs. C")		Cunningham Family
6	Richie	Richard Cunningham ("Richie")		Cunningham Family
7	Joanie	Joanie Cunningham ("Shortcakes")		Cunningham Family
8	Fonzie	Arthur Fonzarelli ("Fonzie", "The Fonz")		
9	Chachi	Charles Arcola ("Chachi")		
10	Ralph	Ralph Malph ("Hector")		
11	Potsie	Warren Weber ("Potsie")		
12				

Press the "Refresh Graph" button and the relationship graph now appears as:



Excel to Graphviz for Win OS

This change has made it very easy to see who the members of the Cunningham family are. Without repeating the details, let's create two more node styles called "Fonzarelli Family" and "Friends". Their styles look as follows:

Style	Format	Preview
Fonzarelli Family	<pre>shape="rect" color="red" fillcolor="lightpink" fixedsize="True" fontcolor="black" fontname="Arial" fontsize="10" style="rounded,filled" height="0.50" width="2.00"</pre>	
Friends	<pre>shape="rect" color="darkorange" fillcolor="lemonchiffon" fixedsize="True" fontcolor="black" fontname="Arial" fontsize="10" style="rounded,filled" height="0.50" width="2.00"</pre>	

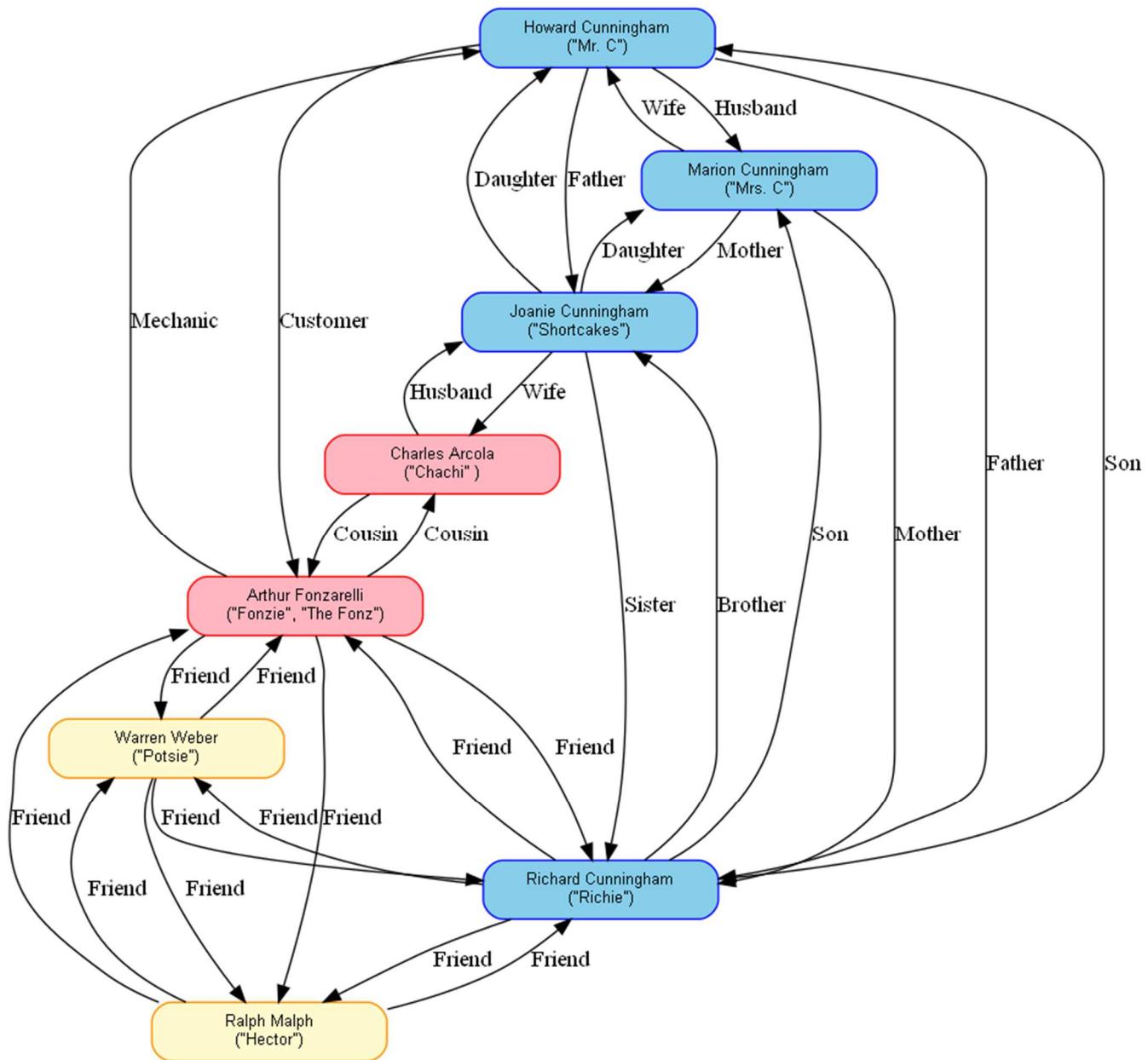
After creating new rows, pasting the style definitions into the 'styles' worksheet, and filling in the other row cells the completed 'styles' worksheet with all 3 styles looks as follows:

A	B	C	D	E	F
	Style Name	Format	Style Type	All Styles	No Clusters
1					
42	Small Circle	shape="circle" height="0.25" fixedsize="True" style="filled" penwidth="1" fontname="Arial" fontsize="8"	node	yes	yes
43	Stop	shape="octagon" height="0.75" width="0.75" fixedsize="True" color="Red" fillcolor="white" fontname="Calibri" fontsize="10" style="filled"	node	yes	yes
44	Cunningham Family	shape="rect" height="0.5" width="2" fixedsize="True" fillcolor="SkyBlue" color="Blue" fontname="Arial" fontsize="10" fontcolor="Black" style="rounded,filled"	node	yes	yes
45	Fonzarelli Family	shape="rect" height="0.5" width="2" fixedsize="True" fillcolor="LightPink" color="Red" fontname="Arial" fontsize="10" fontcolor="Black" style="rounded,filled"	node	yes	yes
46	Friends	shape="rect" height="0.5" width="2" fixedsize="True" fillcolor="LemonChiffon" color="DarkOrange" fontname="Arial" fontsize="10" fontcolor="Black" style="rounded,filled"	node	yes	yes

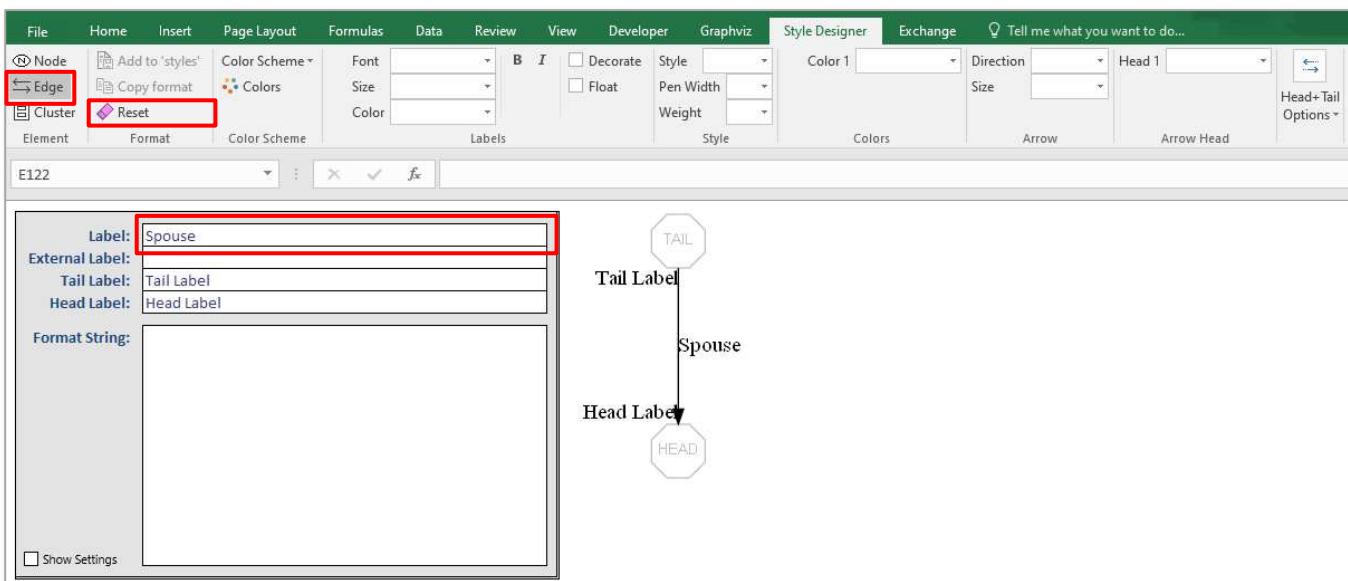
Returning to the 'data' worksheet, Arthur Fonzarelli and Charles Arcola are members of the Fonzarelli Family so associate that style with their names. Ralph Malph, and Warren Weber are friends of the Cunningham and the Fonzarelli families, so associate the "Friends" style with them. The 'data' worksheet now appears as:

A	B	D	G	H
1	Item	Label	Related Item	Style Name
3	Howard	Howard Cunningham ("Mr. C")		Cunningham Family
4	Marion	Marion Cunningham ("Mrs. C")		Cunningham Family
5	Richie	Richard Cunningham ("Richie")		Cunningham Family
6	Joanie	Joanie Cunningham ("Shortcakes")		Cunningham Family
7	Fonzie	Arthur Fonzarelli ("Fonzie", "The Fonz")		Fonzarelli Family
8	Chachi	Charles Arcola ("Chachi")		Fonzarelli Family
9	Ralph	Ralph Malph ("Hector")		Friends
10	Potsie	Warren Weber ("Potsie")		Friends
11				

Press the "Refresh Graph" button and the relationship graph now appears as:



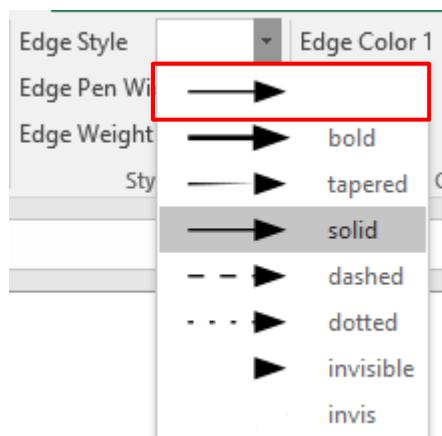
Edges can have styles just as we have seen for nodes. If we return to the 'style designer' worksheet and change the design mode Element to "Edge", we can create edge style definitions using the same method we used for nodes. Press the "Reset Style Designer" button to clear all style values carried over from the node definitions and change the Label Text to "Spouse".



Let us create edge relationships definitions for the relationships "spouse", "parent/child", "friend" and "business associate".

For the "Spouse" definition:

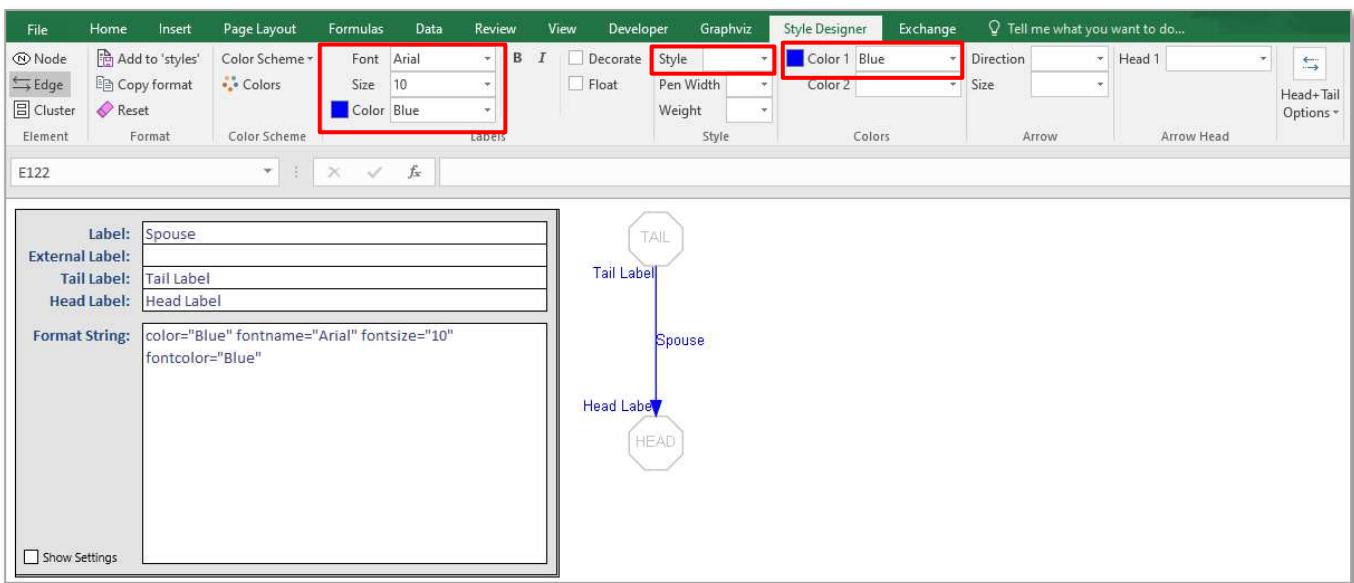
- Set "Edge Style" to "blank"



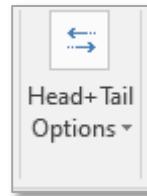
- Set "Edge Color 1" to "blue"
- Set "Edge Font Name:" to "Arial"
- Set "Edge Font Size:" to "10" points
- Set "Edge Font Color:" to "blue"

Excel to Graphviz for Win OS

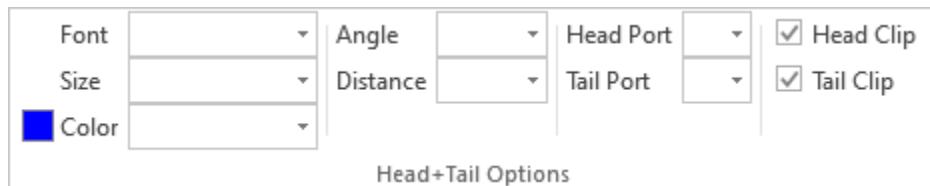
The preview image of the 'edge' now appears as:



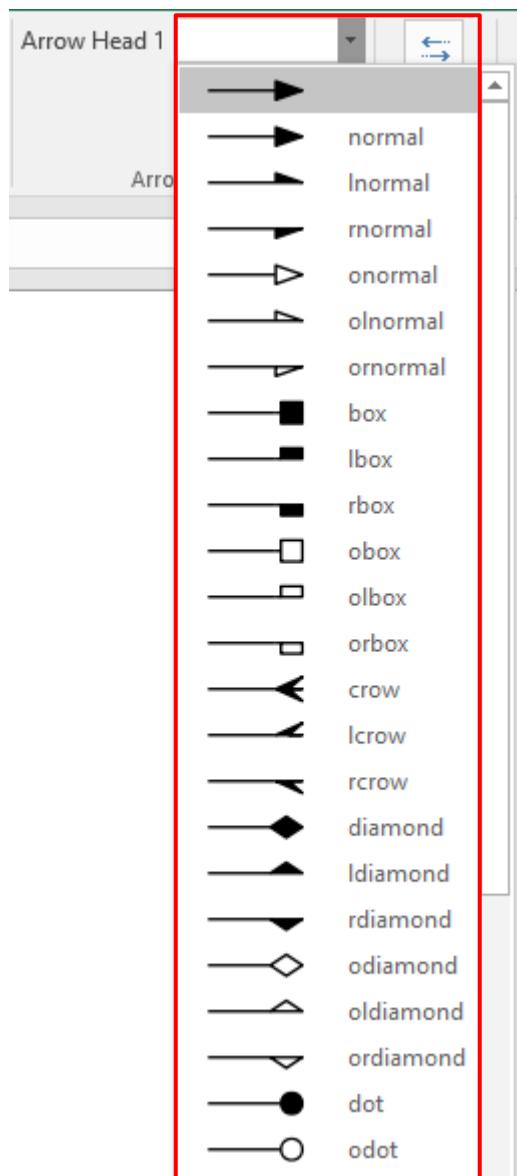
You will notice there are values of "Tail Label" and "Head Label" present in the preview image.



You can specify different fonts, colors and other options by choosing the 'Head+Tail Options'. Notice these options may appear below the ribbon. That behavior occurs when the ribbon is wider than the window size, so a pane is presented below the ribbon.



Arrowheads are another popular option for edges, and Graphviz provides a robust set of choices. You may stack arrowhead choices to build custom arrowheads. The Relationship Visualizer ribbon provides up to 3 arrowheads. When you specify the first arrowhead, a new dropdown list will appear. Likewise, after specifying a second arrowhead, a 3rd dropdown list will appear.



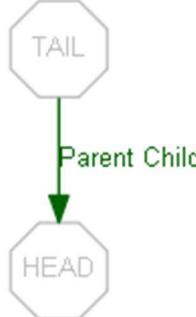
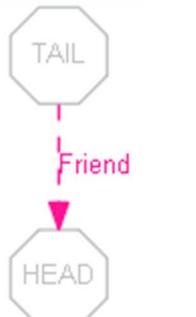
Each change generates the "Edge" Format' string and renders a sample graph showing how the edge will look based upon the *Layout Engine* and *Splines* settings on the 'settings' worksheet. Note that the appearance may change due to how the layout engines render splines, head ports, and tail ports. For more information on these settings, see section **Graph Options** beginning on page 227.

When the edge style is the way you want it to look, pressing the 'Add format to 'styles'' button copies the format string to 'styles' worksheet. Assign the style the name "Spouse".

The 'styles' worksheet should now look as follows:

A	B	C	D	E	F
	Style Name	Format	Style Type	All Styles	No Clusters
1					
44	Cunningham Family	shape="rect" height="0.5" width="2" fixedsize="True" fillcolor="SkyBlue" color="Blue" fontname="Arial" fontsize="10" fontcolor="Black" style="rounded,filled"	node	yes	yes
45	Fonzarelli Family	shape="rect" height="0.5" width="2" fixedsize="True" fillcolor="LightPink" color="Red" fontname="Arial" fontsize="10" fontcolor="Black" style="rounded,filled"	node	yes	yes
46	Friends	shape="rect" height="0.5" width="2" fixedsize="True" fillcolor="LemonChiffon" color="DarkOrange" fontname="Arial" fontsize="10" fontcolor="Black" style="rounded,filled"	node	yes	yes
47	Spouse	color="Blue" fontname="Arial" fontsize="10"	edge	yes	yes
48					

Without repeating all the details, let us create three more node styles called "Parent Child", "Friend", and "Business Associate". The edge styles definitions and previews look as follows:

Style	Format	Preview
Parent Child	style="solid" color="darkgreen" fontcolor="darkgreen" fontname="Arial" fontsize="10"	
Friend Of	style="dashed" color="deeppink" fontcolor="deeppink" fontname="Arial" fontsize="10"	



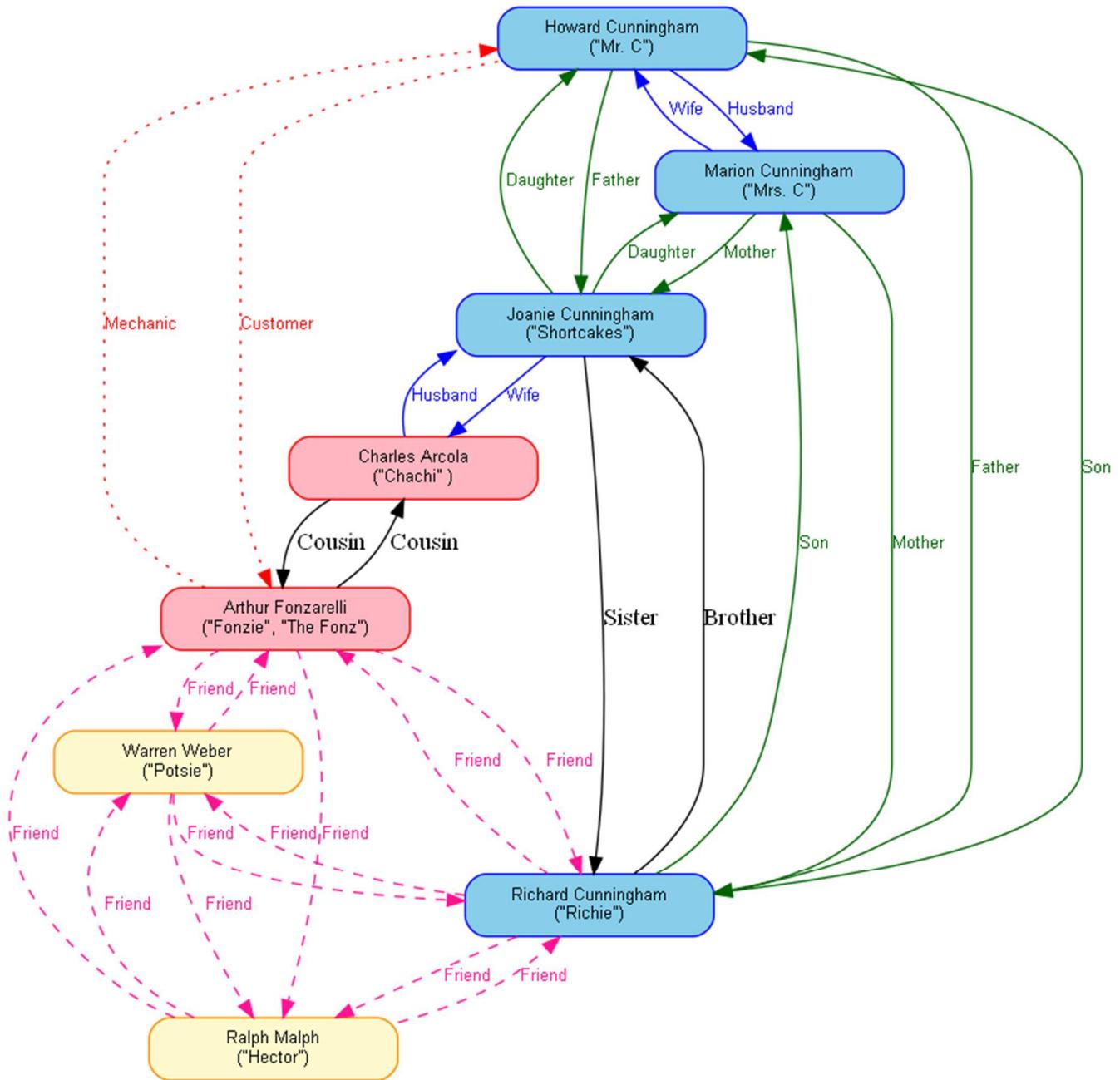
After pasting the style definitions into the 'styles' worksheet, and assigning the style names the completed 'styles' worksheet should now look as follows:

A	B	C	D	E	F
	Style Name	Format	Style Type	All Styles	No Clusters
1					
46	Friends	shape="rect" height="0.5" width="2" fixedsize="True" fillcolor="LemonChiffon" color="DarkOrange" fontname="Arial" fontsize="10" fontcolor="Black" style="rounded,filled"	node	yes	yes
47	Spouse	style="solid" color="Blue" fontname="Arial" fontsize="10" fontcolor="Blue"	edge	yes	yes
48	Parent Child	style="solid" color="DarkGreen" fontname="Arial" fontsize="10" fontcolor="DarkGreen"	edge	yes	yes
49	Friend Of	style="dashed" color="DeepPink" fontname="Arial" fontsize="10" fontcolor="DeepPink"	edge	yes	yes
50	Business Associate	style="dotted" color="Red" fontname="Arial" fontsize="10" fontcolor="Red"	edge	yes	yes

Returning to the 'data' worksheet, assign the appropriate style to the relationship rows based upon the kind of relationship documented. The 'data' worksheet now appears as:

A	B	D	G	H
1	Item	Label	Related Item	Style Name
12				
13	Howard	Husband	Marion	Spouse
14	Howard	Father	Richie	Parent Child
15	Howard	Father	Joanie	Parent Child
16	Howard	Landlord	Fonzie	Business Associate
17	Howard	Customer	Fonzie	Business Associate
18	Marion	Wife	Howard	Spouse
19	Marion	Mother	Richie	Parent Child
20	Marion	Mother	Joanie	Parent Child
21	Richie	Son	Howard	Parent Child
22	Richie	Son	Marion	Parent Child
23	Richie	Brother	Joanie	Sibling
24	Richie	Friend	Fonzie	Friend Of
25	Richie	Friend	Ralph	Friend Of
26	Richie	Friend	Potsie	Friend Of
27	Joanie	Daughter	Howard	Parent Child
28	Joanie	Daughter	Marion	Parent Child
29	Joanie	Sister	Richie	Sibling
30	Joanie	Wife	Chachi	Spouse
31	Fonzie	Friend	Richie	Friend Of
32	Fonzie	Friend	Ralph	Friend Of
33	Fonzie	Friend	Potsie	Friend Of
34	Fonzie	Tenant	Howard	Business Associate
35	Fonzie	Mechanic	Howard	Business Associate
36	Fonzie	Cousin	Chachi	Cousin
37	Chachi	Cousin	Fonzie	Cousin
38	Chachi	Husband	Joanie	Spouse
39	Ralph	Friend	Fonzie	Friend Of
40	Ralph	Friend	Richie	Friend Of
41	Ralph	Friend	Potsie	Friend Of
42	Potsie	Friend	Fonzie	Friend Of
43	Potsie	Friend	Richie	Friend Of
44	Potsie	Friend	Ralph	Friend Of
45				

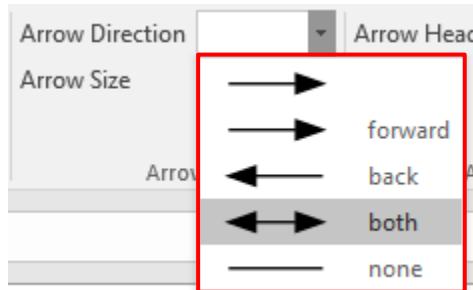
Press the "Refresh Graph" button and the relationship graph now appears as:



Now we begin to visualize the various types of relationships. We can also see that there are still a few rows remaining that do not have styles assigned. These rows represent the "Sister", "Brother", and "Cousin" relationships. These relationships are drawn in the default Graphviz style. We need to go back and create a "Sibling", and a "Cousins" edge style. Using the previous method, define two additional edge styles to look as follows:

Style	Format	Preview
Sibling	style="solid" color="purple" fontcolor="purple" fontname="Arial" fontsize="10"	
Cousin	style="solid" color="chocolate4" arrowtail="dot" dir="both" fontcolor="chocolate4" fontname="Arial" fontsize="10"	

Notice that for the "Cousin" style we have added a filled circle to the Arrow tail. To achieve this effect, you must specify the 'Arrow Direction' as 'Both'.

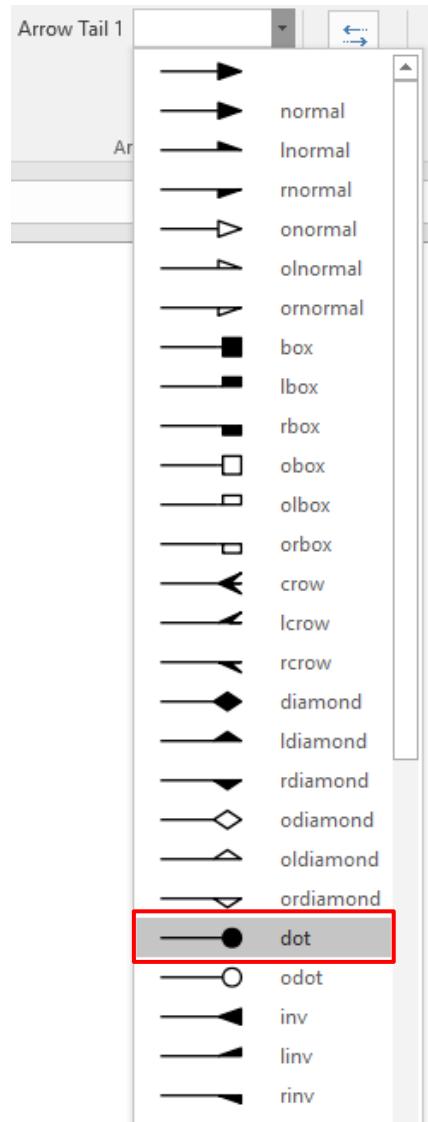


A new set of choices for 'Arrow Tail 1' appear.



Arrow tails have the same set of choices as arrow heads. Like arrow heads, Relationship Visualizer allows you to choose up to 3 styles for Arrow Tails.

Choose "dot" for the filled, rounded tail used in the "Cousin" style.



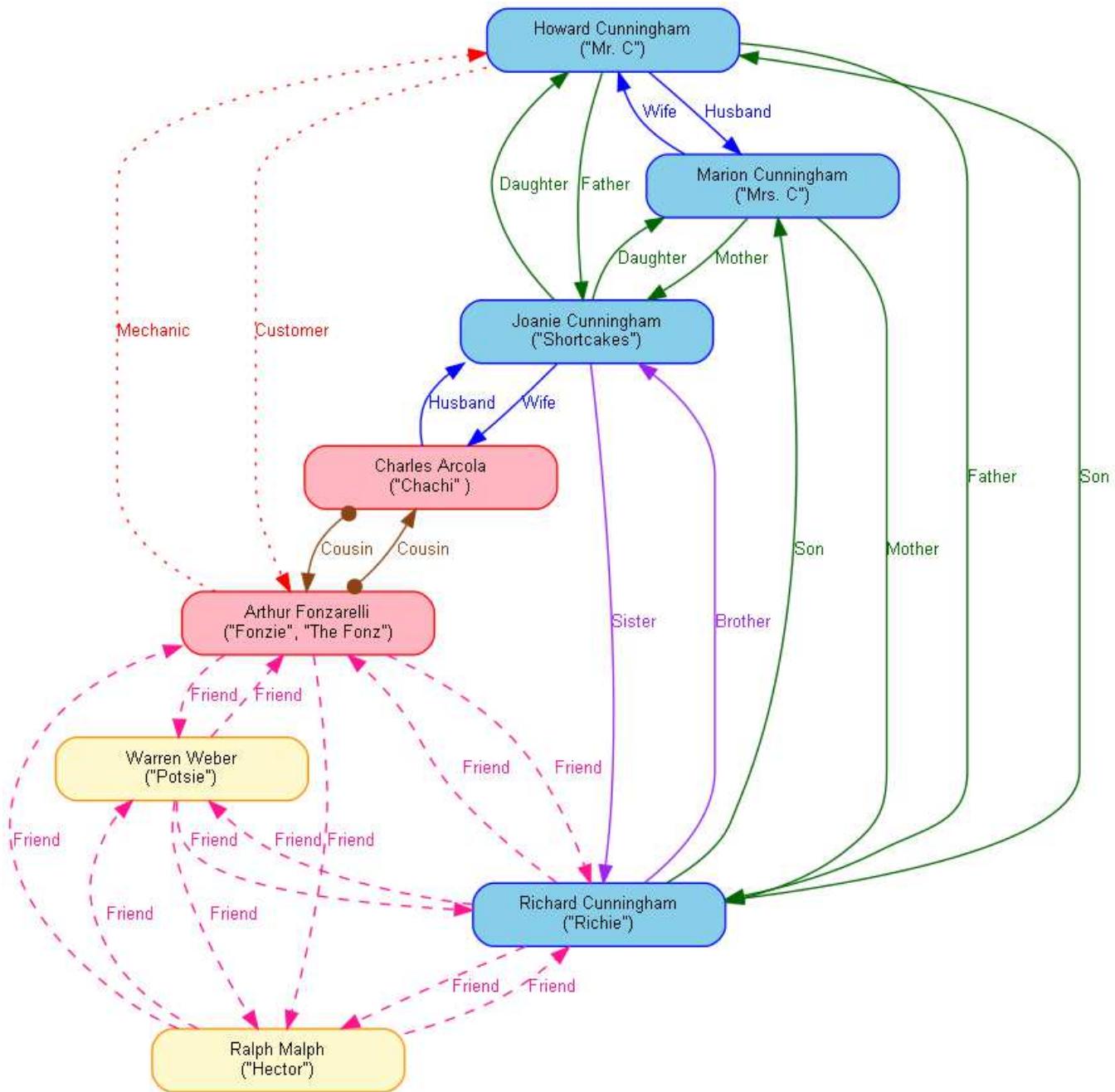
Excel to Graphviz for Win OS

The arrow now appears with the round tail, and the pointed arrow head as shown below.

The screenshot shows the Microsoft Excel ribbon with the 'Graphviz' tab selected. The 'Style Designer' dialog is open, displaying settings for 'Color 1' (Chocolate4) and 'Color 2' (Chocolate4). Under 'Arrow', 'Direction' is set to 'both', 'Head 1' is a dot, 'Tail 1' is a dot, and 'Tail 2' is a dot. The 'Head-Tail Options' dropdown is visible. On the left, a preview pane shows a node labeled 'Spouse' with a tail labeled 'Tail Label' and a head labeled 'Head Label'. Below it is another node labeled 'HEAD'.

Returning to the 'data' worksheet, assign the "Sibling" and "Cousin" style to the remaining rows which currently use the built-in style of "edge" based upon the kind of relationship documented.

Press the "Refresh Graph" button and the relationship graph now appears as:



At this point, it is up to an interested reader to divide the relationships to lower levels and add more styles. It is possible to split the styles we just created down to smaller levels such as Father, Son, Mother, Daughter, Sister, and Brother or even by business relationships such as Landlord and Tenant. There are additional relationships that can be added. For example, when Joanie married Chachi, Howard became Chachi's father-in-law, Marion became Chachi's Mother-in-Law, Chachi became Howard and Marion's Son-in-Law. Add these relationships to expand the model. Have fun with it.

Modifying the Style of Individual Elements

There may be times when you want to embellish the style of a node or an edge. The introduction to this manual described how to determine the route between two states. Perhaps you want to show the shortest route in a different color, make the lines thicker, or use a different style. The Relationship Visualizer has capabilities to support this requirement.

The 'data' worksheet provides a column called "Attributes". Adding additional style or control attributes here will only apply to that row of data. The "Graphviz" ribbon tab controls the "Attributes" capability which can be turned on or off depending upon how you wish to depict the graph.

Display of the "HELP - attributes" worksheet is toggled from the 'Graphviz' ribbon tab ribbon tab. It contains filters on graph elements and layout engine columns which allow you to narrow down the list of attribute you seek. In the illustration below we are filtering on 'edge' attributes, and the 'dot' layout engine:

Attribute	Description	Data Type	Default	Minimum	Notes
arrowhead	Style of arrowhead on the head node of an edge. This will only appear if the dir attribute is "forward" or "both".	arrowType	normal		See limitation.
arrowsize	Multiplicative scale factor for arrowheads.	double	1	0	
arrowtail	Style of arrowhead on the tail node of an edge. This will only appear if the dir attribute is "backward" or "both".	arrowType	normal		See limitation.
color	Basic drawing color for graphics, not text. For the latter, use the fontcolor attribute.	color colorList	black		

For edges, the value can either be a single color or a colorList. In the latter case, if colorList has no fractions, the edge is drawn using parallel splines or lines, one for each color in the list, in the order given. The head arrow, if any, is drawn using the first color in the list, and the tail arrow, if any, the second color. This supports the common case of drawing opposing edges, but using parallel splines instead of separately routed multiedges. If any fraction is used, the colors are drawn in series, with each color being given roughly its specified fraction of the edge. For example, the graph

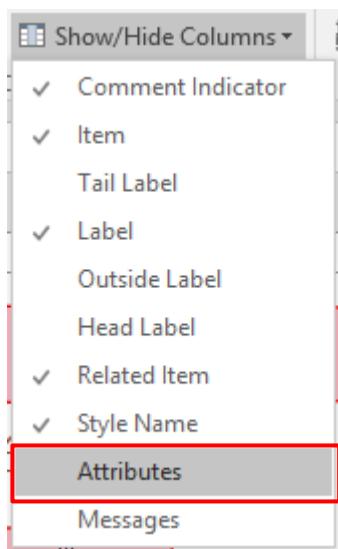
```

graph LR
    a --> b [dir=both color="red:blue"]
    c --> d [dir=none color="green:red:0.25:blue"]
}

```

yields

If the 'Attributes' column is hidden, make it visible by selecting it from the 'Show/Hide Columns' drop-down menu on the 'Graphviz' tab.



The "Happy Days" introduction mentioned that Fonzie emerged as the star of the television show. In this case, it might be useful to use a 'star' shape to identify his node.

Use the 'Style Designer' to define a 2-inch by 2-inch star. Instead of adding the format to the 'styles' worksheet, copy the format to the clipboard.

The screenshot shows the Microsoft Excel ribbon with the 'Style Designer' tab selected. On the far left of the ribbon, there are icons for 'Node', 'Edge', and 'Cluster'. The 'Edge' icon has a red box around its 'Copy format' button. The 'Style Designer' tab has several dropdown menus and input fields: 'Shape' set to 'star', 'Height' and 'Width' both set to '2"', 'Fixed Size' set to 'true', 'Font' set to 'Arial', 'Size' set to '10', and 'Color' set to 'Black'. Below the ribbon, the formula bar shows 'E122'. The main area displays the 'Style Designer' dialog box. It has fields for 'Label' (containing 'Spouse') and 'External Label'. A larger text area labeled 'Format String' contains the code: 'shape="star" height="2" width="2" fixedsize="True" fontname="Arial" fontsize="10" fontcolor="Black"'. At the bottom left of the dialog box is a checkbox labeled 'Show Settings'. To the right of the dialog box is a preview window showing a black five-pointed star with the word 'Spouse' written in the center.

Excel to Graphviz for Win OS

Find the row with Fonzie's name (row 8) and paste the style attributes

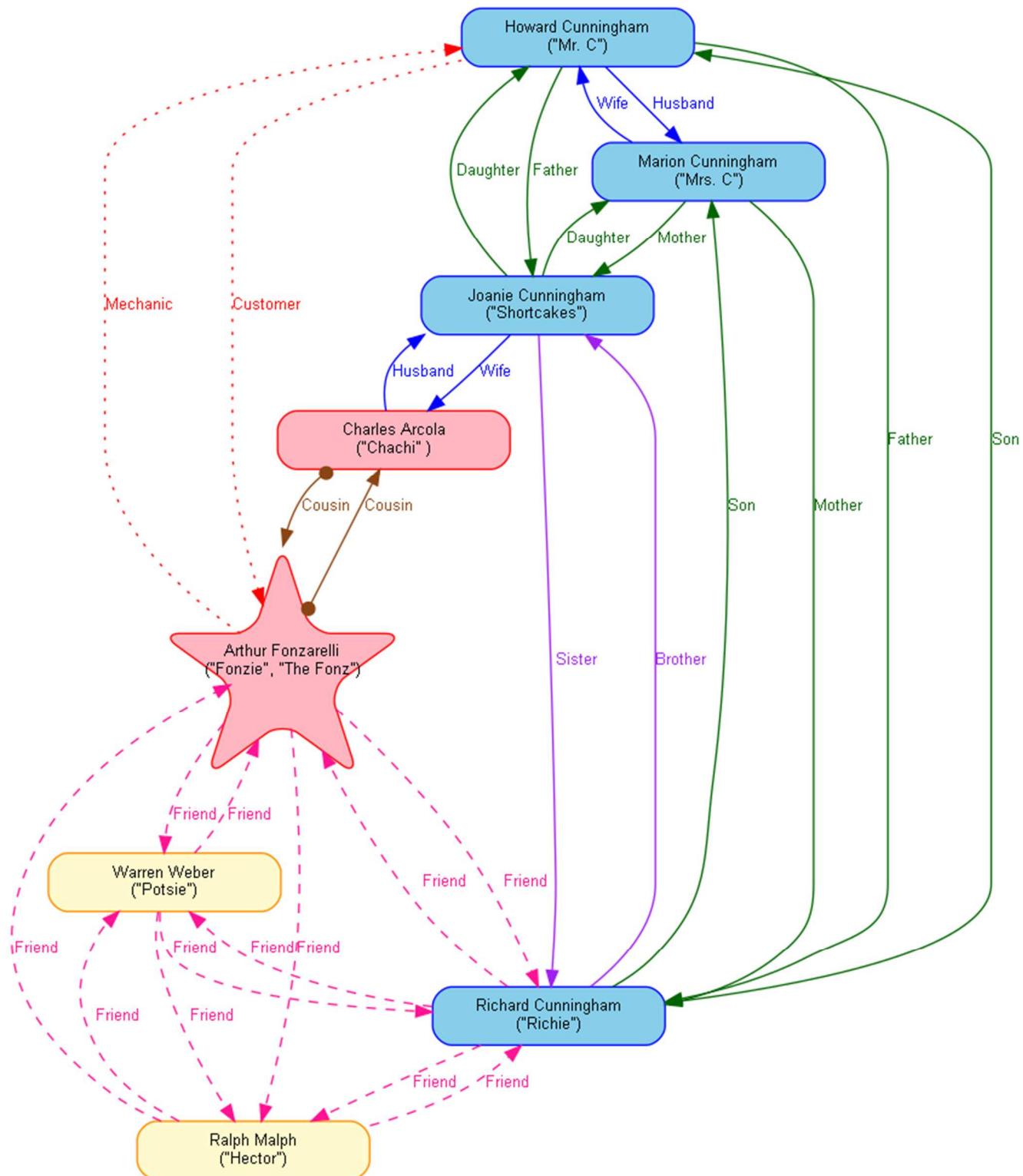
```
shape="star" height="2.0" width="2.0" fixedsize="true"
```

within the "Attributes" column on his row. The height and width are changed to make the star a little larger than the other nodes, and the fixedsize attribute tells Graphviz to make the star exactly 2.0 x 2.0 inches (i.e., do not grow or shrink the star based upon the size of the label).

The spreadsheet appears as:

A	B	D	G	H	I
1	Item	Label	Related Item	Style Name	Attributes
5	Marion	Marion Cunningham ("Mrs. C")		Cunningham Family	
6	Richie	Richard Cunningham ("Richie")		Cunningham Family	
7	Joanie	Joanie Cunningham ("Shortcakes")		Cunningham Family	
8	Fonzie	Arthur Fonzarelli ("Fonzie", "The Fonz")		Fonzarelli Family	<pre>shape="star" height="2" width="2" fixedsize="True" fontname="Arial" fontsize="10" fontcolor="Black"</pre>
	Chachi	Charles Arcola ("Chachi")		Fonzarelli Family	

Press the "Refresh Graph" button and the relationship graph now appears as:



Notice that the "Attributes" attributes are additive. The star has rounded points because the "Fonzarelli Family" style associated with Arthur Fonzarelli has an attribute which includes `style="rounded"`, and the border and fill colors remain unchanged.

Grouping Information in Clusters

The DOT layout engine supports the cluster concept described previously in this manual. Other layout engines such as FDP and SFDP do not. This means that some of the style techniques described here will not work if you are not using the DOT layout engine.

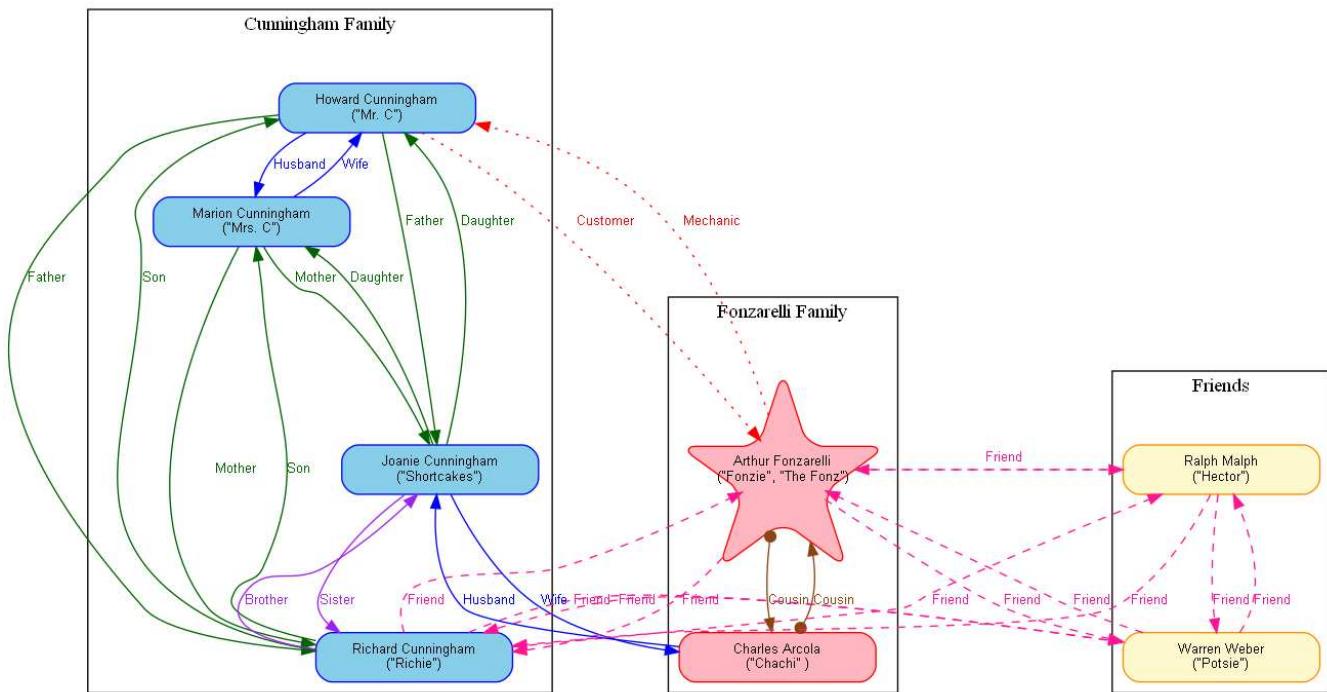
The Relationship Visualizer allows you to denote the start of a cluster with open brace "{" and close brace "}" characters in the "Item" column. The "Label", "Style", and "Attributes" columns all work for clusters as was previously described for nodes and edges.

In the previous graph Richie Cunningham is located at the bottom of the graph, while the other members of his family are at the top of the graph. We would like to group all the members of the Cunningham family together, so let's add an open brace "{" above their rows with a label of "Cunningham Family", and a close brace "}" below the row of the last Cunningham family member. We will also group the Fonzarelli family members and group the Friends. To make the data more readable, we will also add comments by placing a "#" character in Column A, and comment text in the Item cells

The spreadsheet now appears as:

	A	B	D	G	H	I
1	Item	Label	Related Item	Style Name	Attributes	
2	# Cunningham Family					
3	{	Cunningham Family				
4	Howard	Howard Cunningham ("Mr. C")		Cunningham Family		
5	Marion	Marion Cunningham ("Mrs. C")		Cunningham Family		
6	Richie	Richard Cunningham ("Richie")		Cunningham Family		
7	Joanie	Joanie Cunningham ("Shortcakes")		Cunningham Family		
8	}					
9	# Fonzarelli Family					
10	{	Fonzarelli Family		Fonzarelli Family	shape="star" height="2" width="2" fixedsize="True" fontname="Arial" fontsize="10" fontcolor="Black"	
11	Fonzie	Arthur Fonzarelli ("Fonzie", "The Fonz")				
12	Chachi	Charles Arcola ("Chachi")		Fonzarelli Family		
13	}					
14	# Friends					
15	{	Friends		Friends		
16	Ralph	Ralph Malph ("Hector")				
17	Potsie	Warren Weber ("Potsie")		Friends		
18	}					

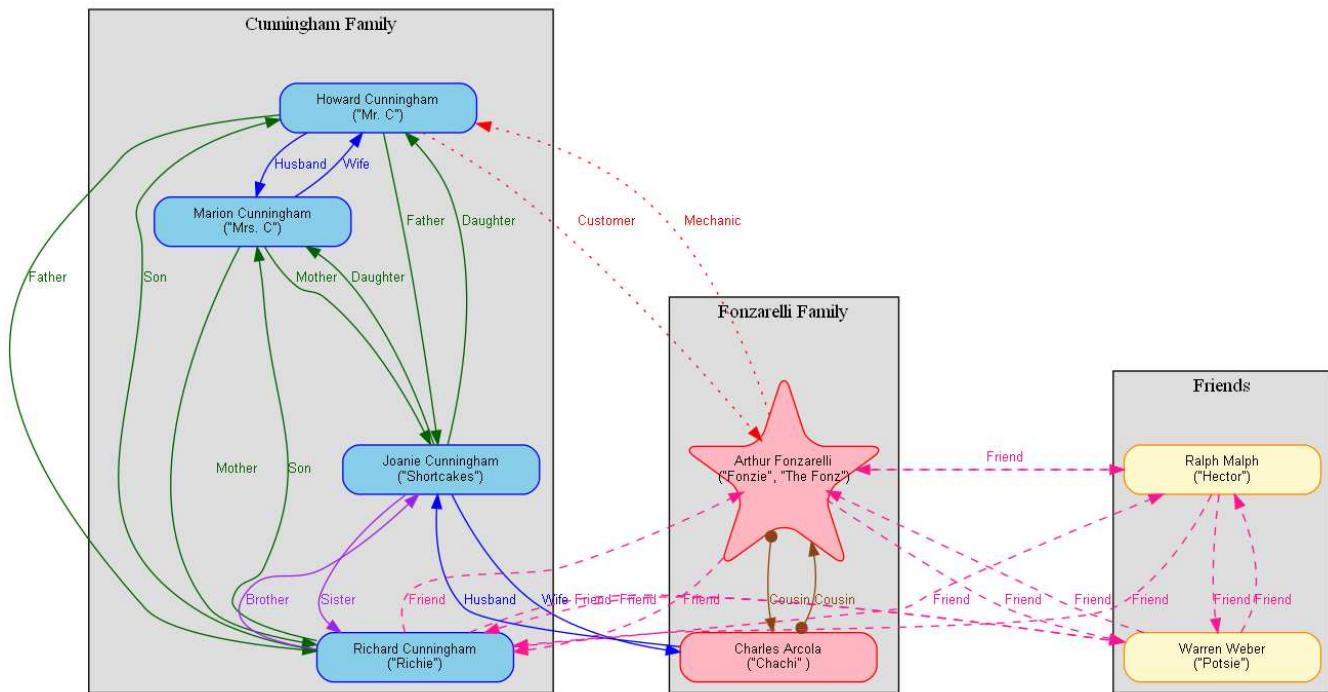
Press the "Refresh Graph" button and the relationship graph now appears as:



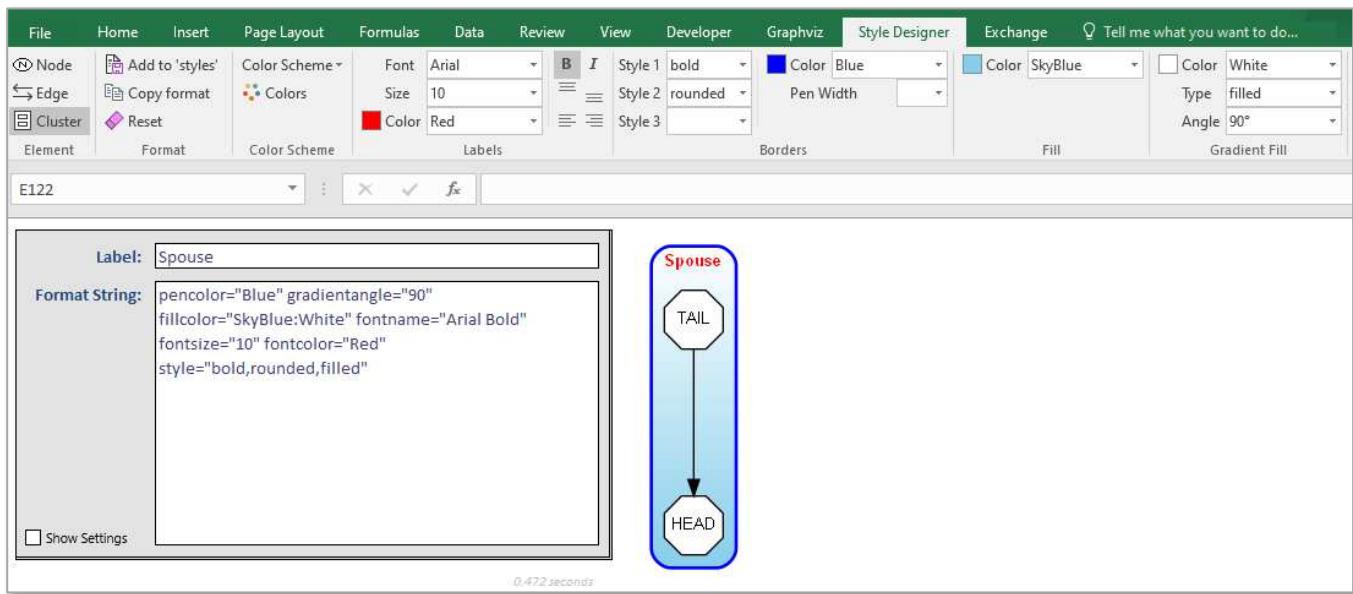
We can use the "Attributes" capabilities for clusters, just as we did for nodes. For example, add the attribute "bgcolor="gray87" in the "Attributes" column on the three rows containing the open cluster "{" characters:

Item	Label	Is Related To Item	Style Name	Extra Style Attributes
{	Cunningham Family			bgcolor="gray87"

Press the "Refresh Graph" button and the relationship graph now appears as:



The 'style designer' worksheet contains a "Cluster" design mode for creating Cluster style definitions. It works the same way as was explained for nodes and edges, and therefore an additional explanation here would probably bore the reader. The following picture provides an example of a "Cluster" style definition.



Adding Native Graphviz Directives

The Relationship Visualizer is intended to hide as much of Graphviz's DOT language as possible so that you only manipulate rows and columns in Excel. You can take greater control of the diagram's presentation by adding in Graphviz statements, assuming you take the time to learn the DOT language².

You can insert Graphviz commands into the DOT language file by specifying the character ">" in the Item column. The ">" character identifies the row as having a "native" command. When an item of ">" is encountered the Excel macro code takes the value in the "Label" column and writes it verbatim to the text file which will be sent to the layout engine.

Note: Labels are processed to convert Excel-style carriage returns into Graphviz-style carriage returns, as well as filter out special characters that Graphviz cannot process. Rows with native Graphviz commands are not inspected, and the value in the Label column is written directly as-is into the Graphviz file.

For example, the Graphviz statements which will depict Howard and Marion Cunningham on a common plane, and Richie and Joanie Cunningham on a separate, yet common plane are:

```
{rank="same"; "Howard"; "Marion";}
{rank="same"; "Richie"; "Joanie";}
```

These directives must be placed within the cluster where the Cunningham family is defined.

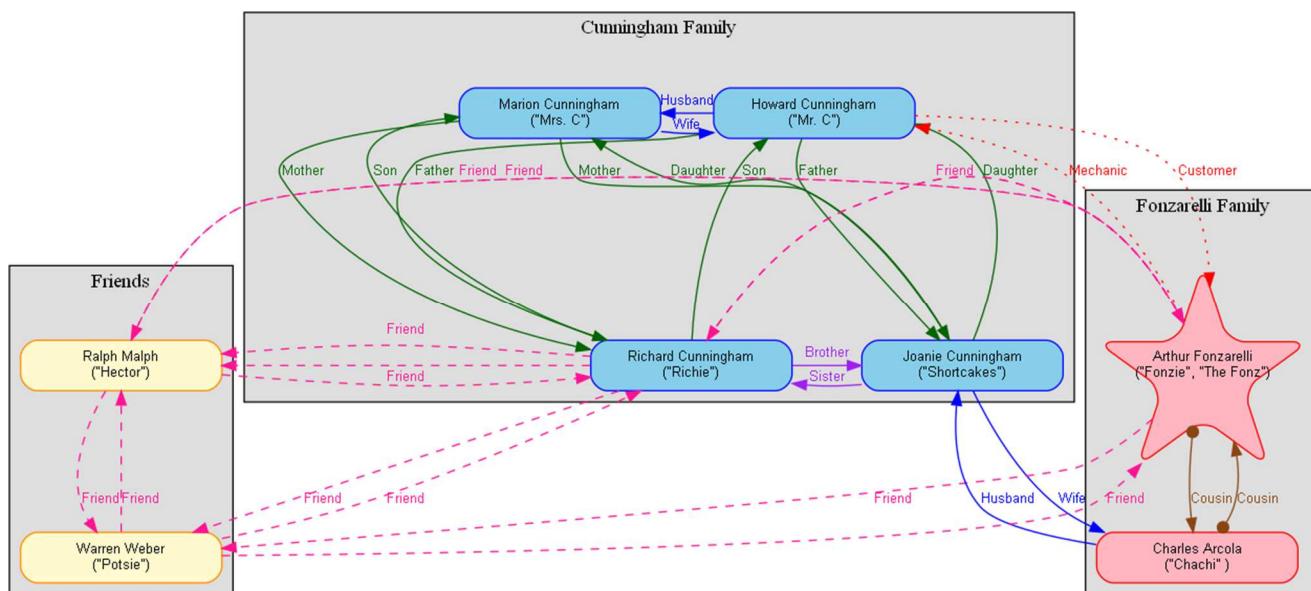
Add two rows above the close brace "}". In the "Item" column enter the character ">", and in the "Label" column enter the Graphviz statements listed above.

² Readers interested in learning more about the DOT language should refer to <https://graphviz.org/documentation/> and <http://www.graphviz.org/pdf/dotguide.pdf>

The Excel data now appears as:

A	B	D	G	H	I
1	Item	Label	Related Item	Style Name	Attributes
2	# Cunningham Family				
3	{	Howard Cunningham Family		Cunningham Family	bgcolor="gray87"
4	Howard	Howard Cunningham ("Mr. C")		Cunningham Family	
5	Marion	Marion Cunningham ("Mrs. C")		Cunningham Family	
6	Richie	Richard Cunningham ("Richie")		Cunningham Family	
7	Joanie	Joanie Cunningham ("Shortcakes")		Cunningham Family	
8	>	{rank="same"; "Howard"; "Marion";}			
9	>	{rank="same"; "Richie"; "Joanie";}			
10	}				
11	# Fonzarelli Family				

Press the "Refresh Graph" button and the relationship graph now appears as:

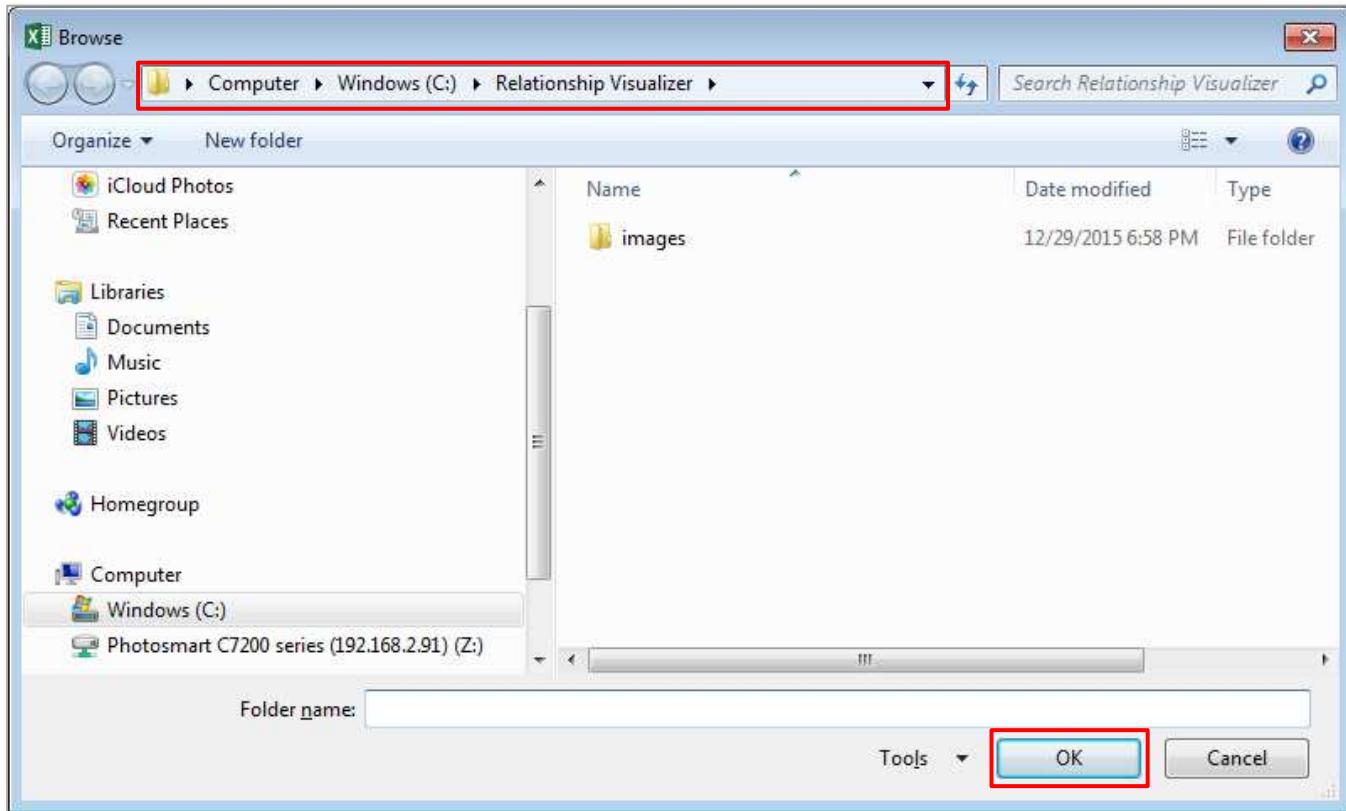


Adding Images

As you develop more advanced relationship graphs you may want to use images to represent the nodes in combination with, or in place of the node shapes. Graphviz supports an "image=" attribute where you can provide a file name of an image to include in a node.

The Relationship Visualizer by default will look in the directory where the spreadsheet is saved. If you wish to store the images in other locations, you must make a configuration change on the 'settings' worksheet to specify the location(s). The image path must be specified before you can use the "image=" attribute in a style definition.

Switch to the 'settings' worksheet and locate the "Image Path:" setting in the 'Graph Options' section. To the right of the cell is a button with three dots [...]. If you press that button it will bring up the standard directory selection dialog which you can use to choose the directory where the images are stored. Navigate to the directory and press the "OK" button to transfer the path to the cell.



Your settings should appear like this:

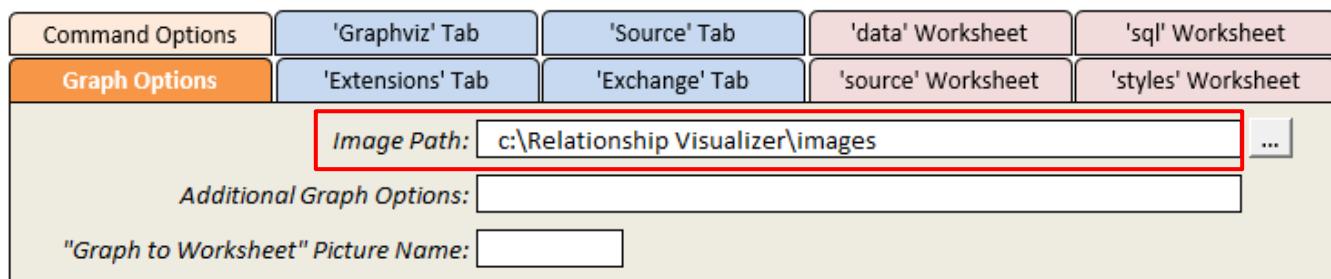
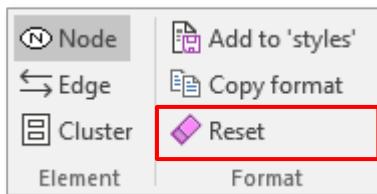


Image name is an option on the 'style designer' worksheet that is useful when you want to create a common style definition where all nodes of a given style use a common icon. For example, it is possible to depict computers with one image, depict databases with another image, and depict computer programmers with yet another image.

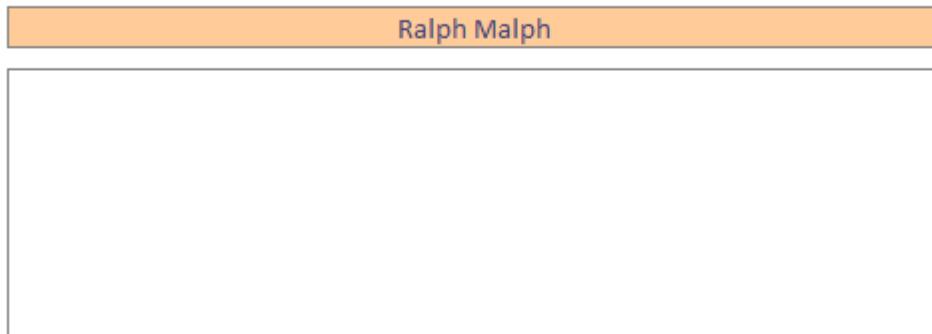
For our example, we are going to take it a step further by using the "Attributes" column to insert a photograph of each character.

The characters of Potsie and Ralph have not gotten very much attention in this example, so let us shift the focus to them. We have obtained JPEG images of these characters off the internet, cropped them to be square, scaled them down to a common size, and placed them in the directory where the Excel workbook is saved.

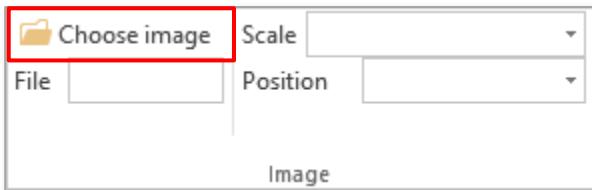
Step 1 - Press the "Reset style designer" button to clear the format string and reset the ribbon controls.



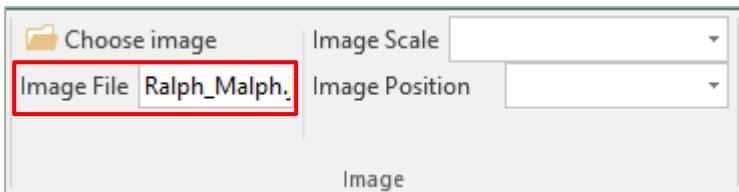
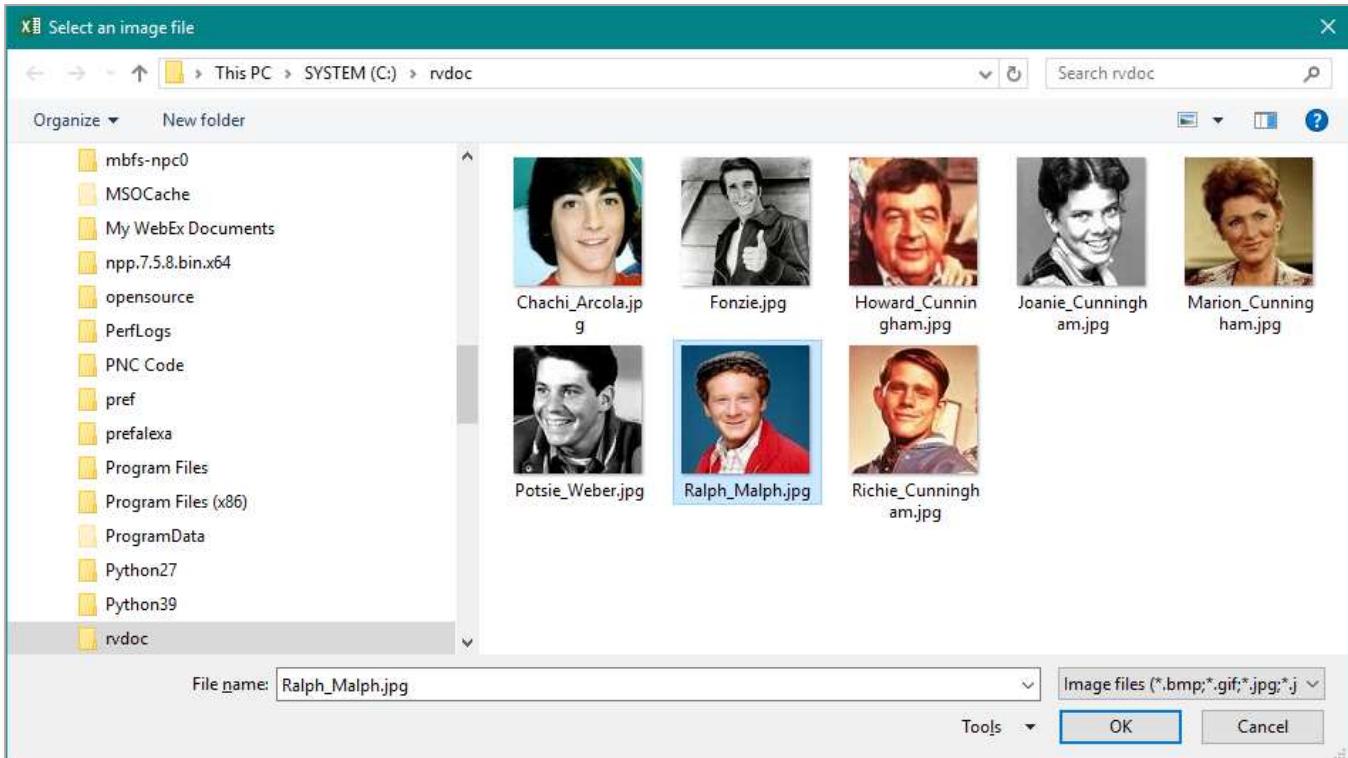
Label Text:



Step 2 - Press the "Choose Image" button in the Image section of the tab



Step 3 - Navigate to the directory containing the images and choose Ralph Malph's picture.



Label Text:

Ralph Malph

image="Ralph_Malph.jpg"



Step 4 - We want the image aspect ratio maintained so that the image does not get stretched vertically or horizontally. Choose "Keep aspect ratio" from the Image Scale dropdown list.

Choose image

Image File **Ralph_Malph.**

Image Scale **Keep aspect ratio**

Image Position

Natural image size

Scale to node width

Scale to node height

Scale both separately

Label Text:

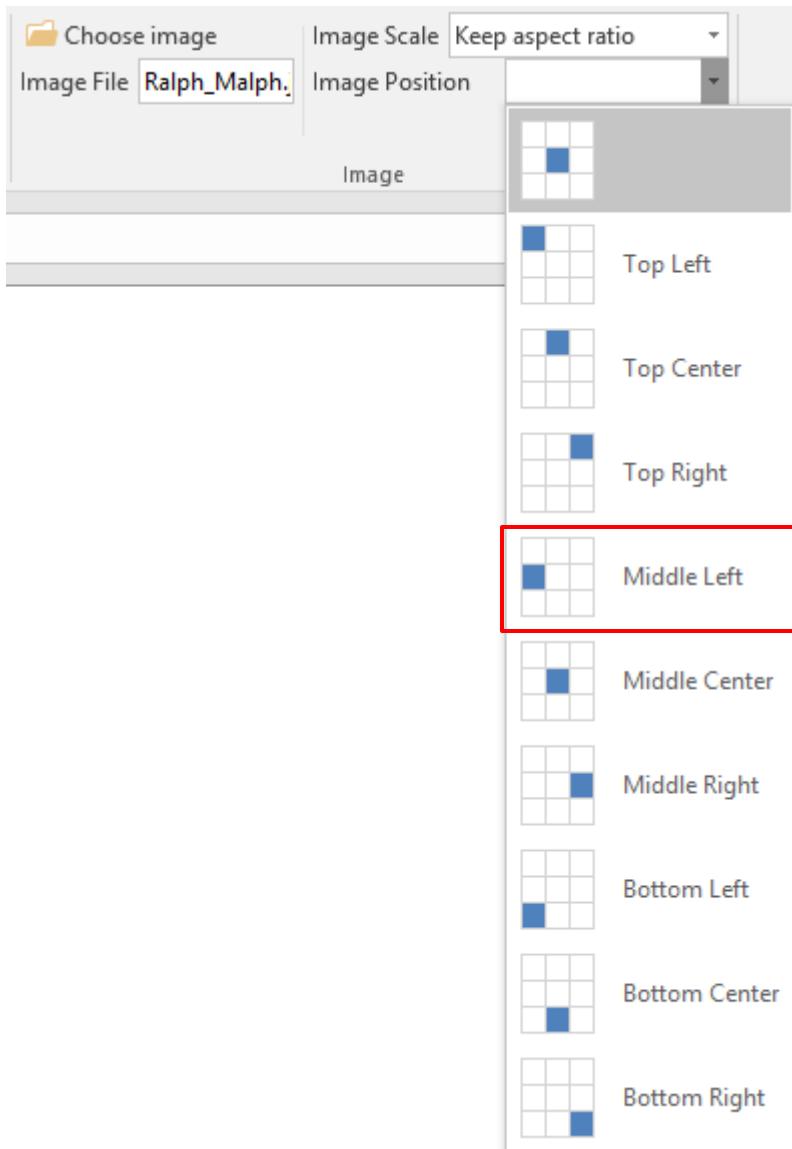
Ralph Malph

image="Ralph_Malph.jpg" imagescale="True"



Note that the image is full sized, and centered in the default oval shape.

Step 5 - We want to shift the image position to the left edge so that it is placed beside the character's name. Select "Middle Left" from the Image Position dropdown list.



Label Text:

Ralph Malph

```
image="Ralph_Malph.jpg" imagescale="True" imagepos="ml"
```



Step 6 - At this point you may be thinking "What about the shape and font, they are not right?" We don't worry about them because this attribute string will be pasted into the "Attribute" and will be added to, and potentially override any Graphviz attribute defined by the saved style specified in the "Style Name" column.

For Ralph, we will add Attributes attributes:

```
image="Ralph_Malph.jpg" imagescale="True" imagepos="ml"
```

Label Text:

```
Ralph Malph
```



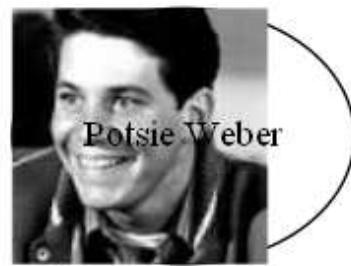
```
image="Ralph_Malph.jpg" imagescale="True" imagepos="ml"
```

and for Potsie we will add:

```
image="Potsie_Weber.jpg" imagescale="True" imagepos="ml"
```

Label Text:

```
Potsie Weber
```



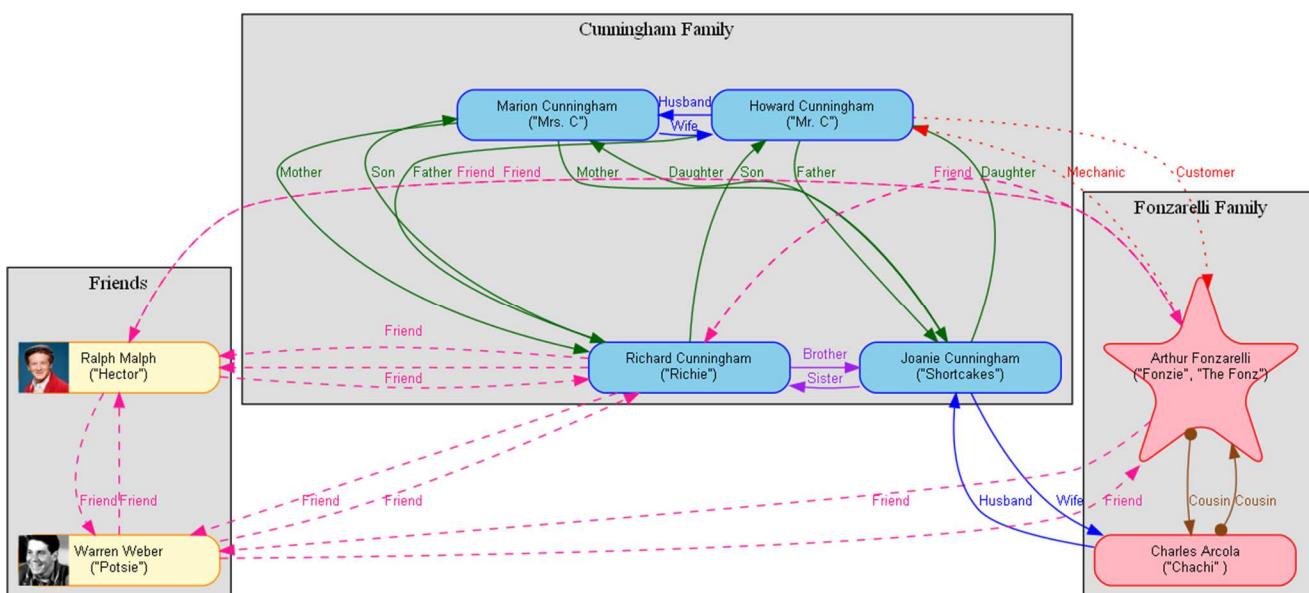
```
image="Potsie_Weber.jpg" imagescale="True" imagepos="ml"
```

The `imagepos` attribute will move the label location from the default center position to the middle left of the shape. The `imagescale` attribute will scale the image down to fit within the shape, while maintaining the aspect ratio.

The Excel data now appears as follows:

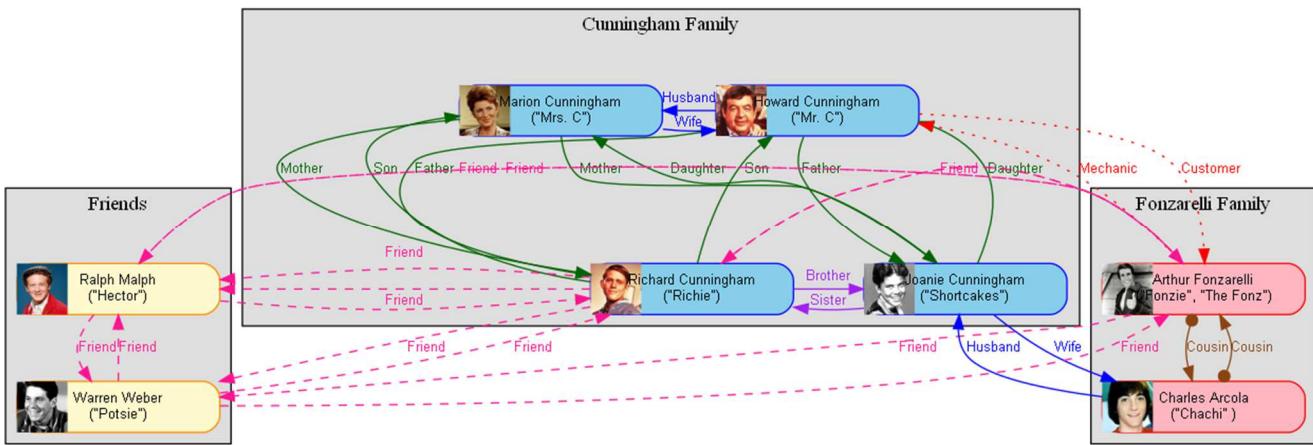
A	B	D	G	H	I
1	Item	Label	Related Item	Style Name	Attributes
16	# Friends				
17	{	Friends			bgcolor="gray87"
18	Ralph	Ralph Malph ("Hector")		Friends	image="Ralph_Malph.jpg" imagescale="true" imagepos="ml"
19	Potsie	Warren Weber ("Potsie")		Friends	image="Potsie_Weber.jpg" imagescale="True" imagepos="ml"
20	}				

Press the "Refresh Graph" button and the relationship graph now appears as:



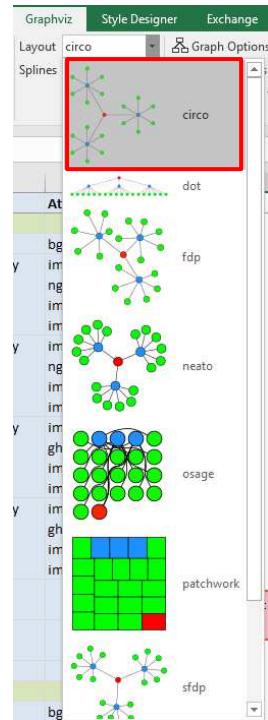
We now see the graph contains the images specified on the two lines. The shape is still rectangular of fixed height and width, with a light-yellow background and orange border as dictated by the "Friends" style. The image attributes have been added to the style definition. The images have been scaled down to fit the height of the shape and are aligned with the left edge of the rectangle.

Step 7 - Next, we repeat these steps for the other characters (and remove shape="star" for Fonzie). The graph now appears as follows:

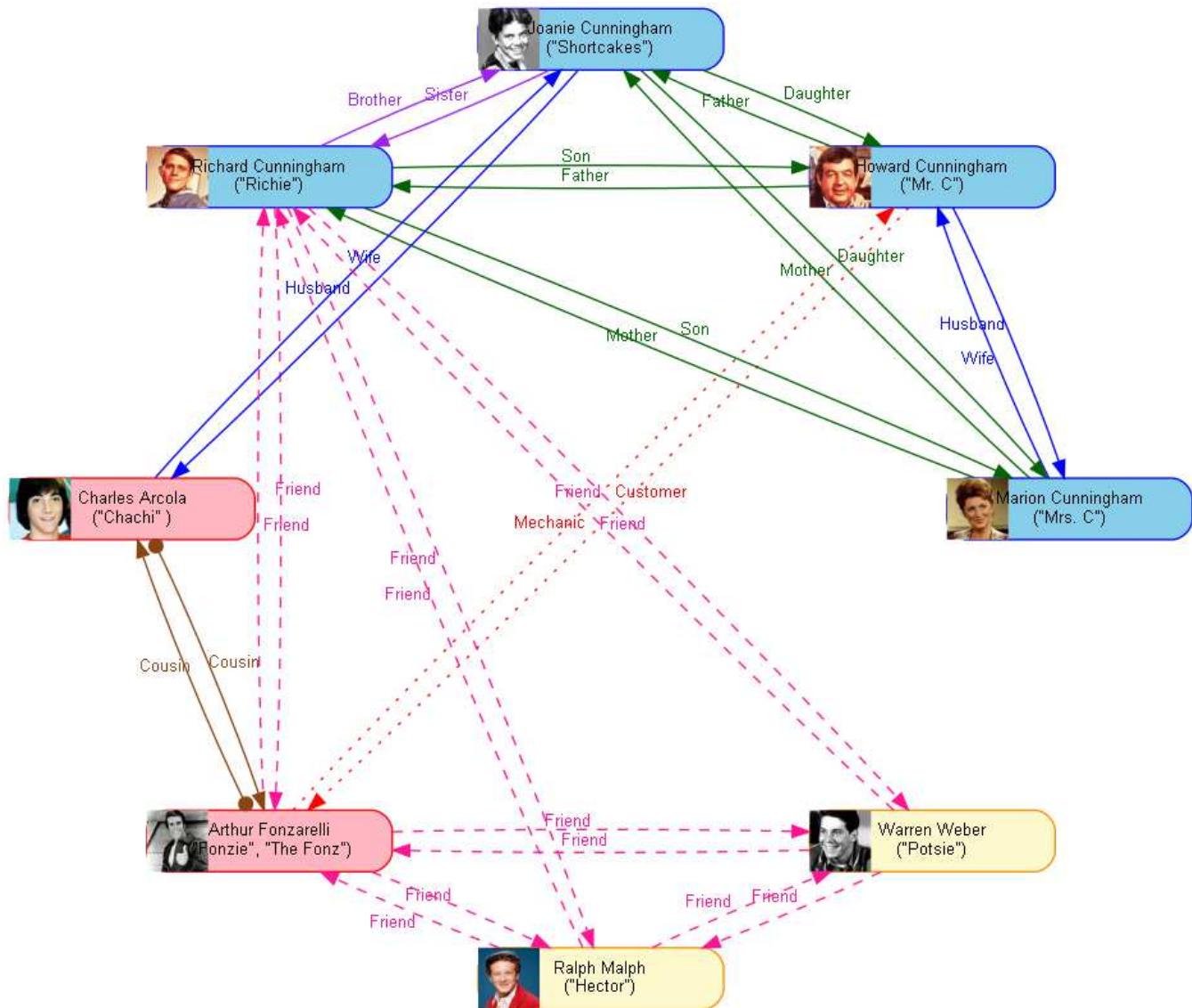


We can make additional changes to the graph by trying different Graphviz graph attributes. For example, Graphviz has two algorithms for ranking nodes when clusters are used.

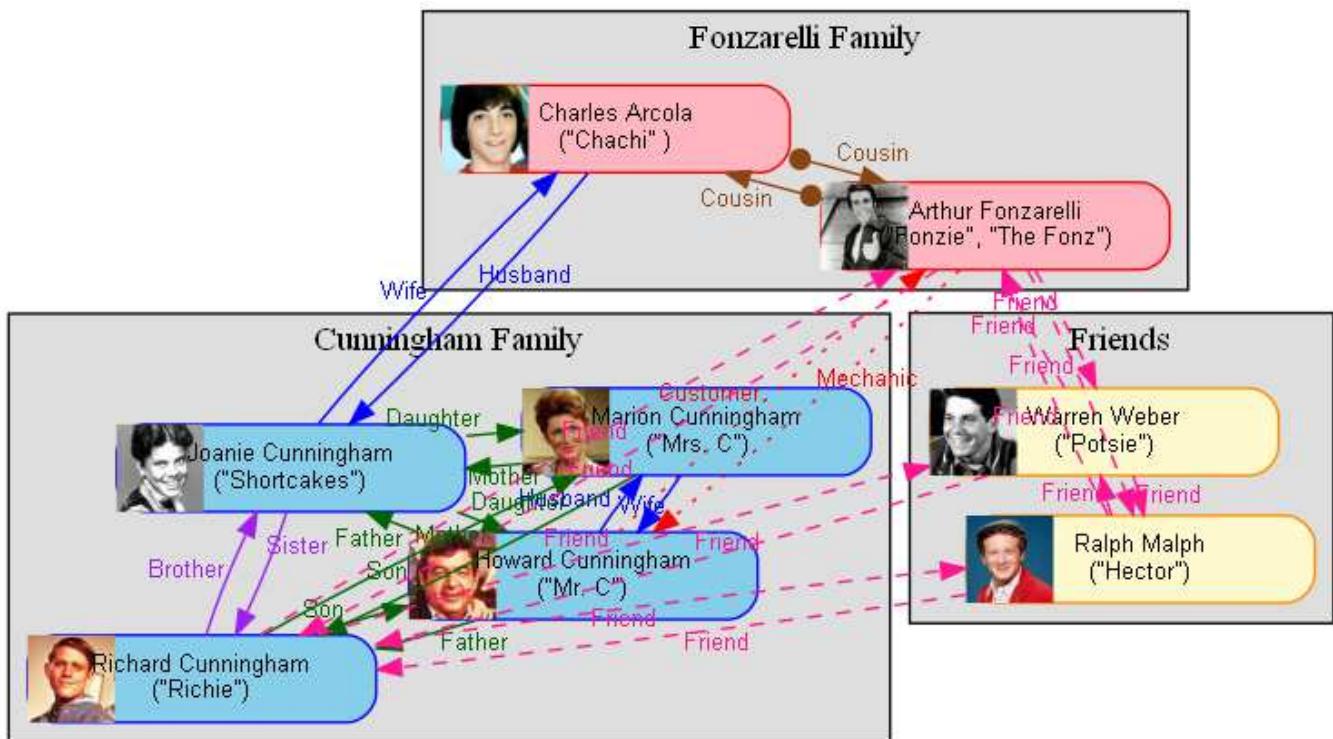
We can play with some of the other Graphviz options now to see different ways the graph can be displayed. If we choose a different layout algorithm, let's see what happens to the graph. Choose "circo" from the Layout dropdown:



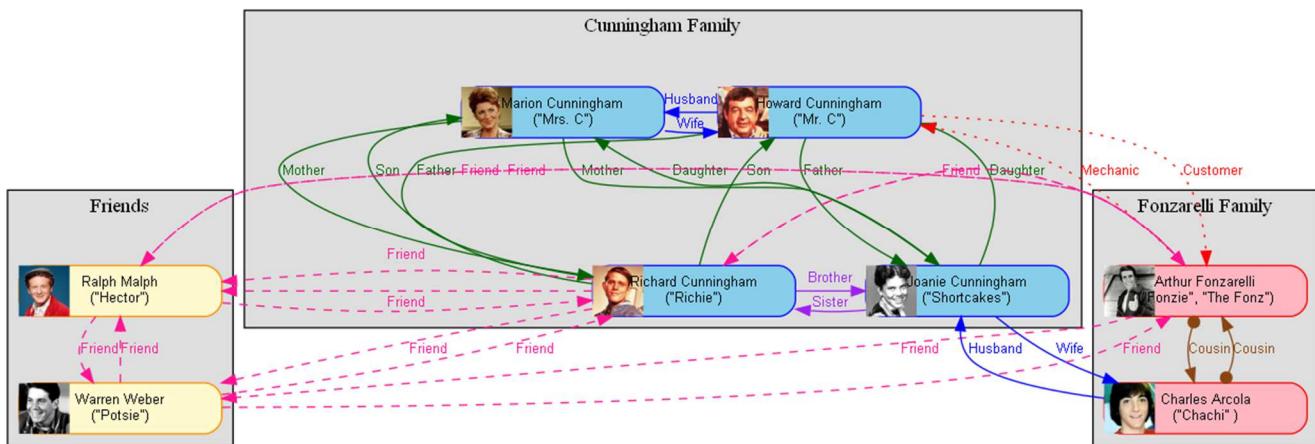
The graph now appears as:



Notice that the cluster are gone, as "circo" does not support clusters. Now repeat, but choose algorithm "fdp" which does support clusters. The graph now appears as:

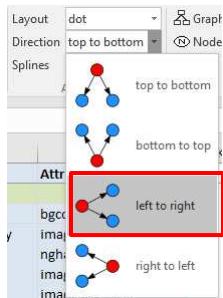


Now return to the original layout by selecting the "dot" algorithm. Once again the diagram appears as:

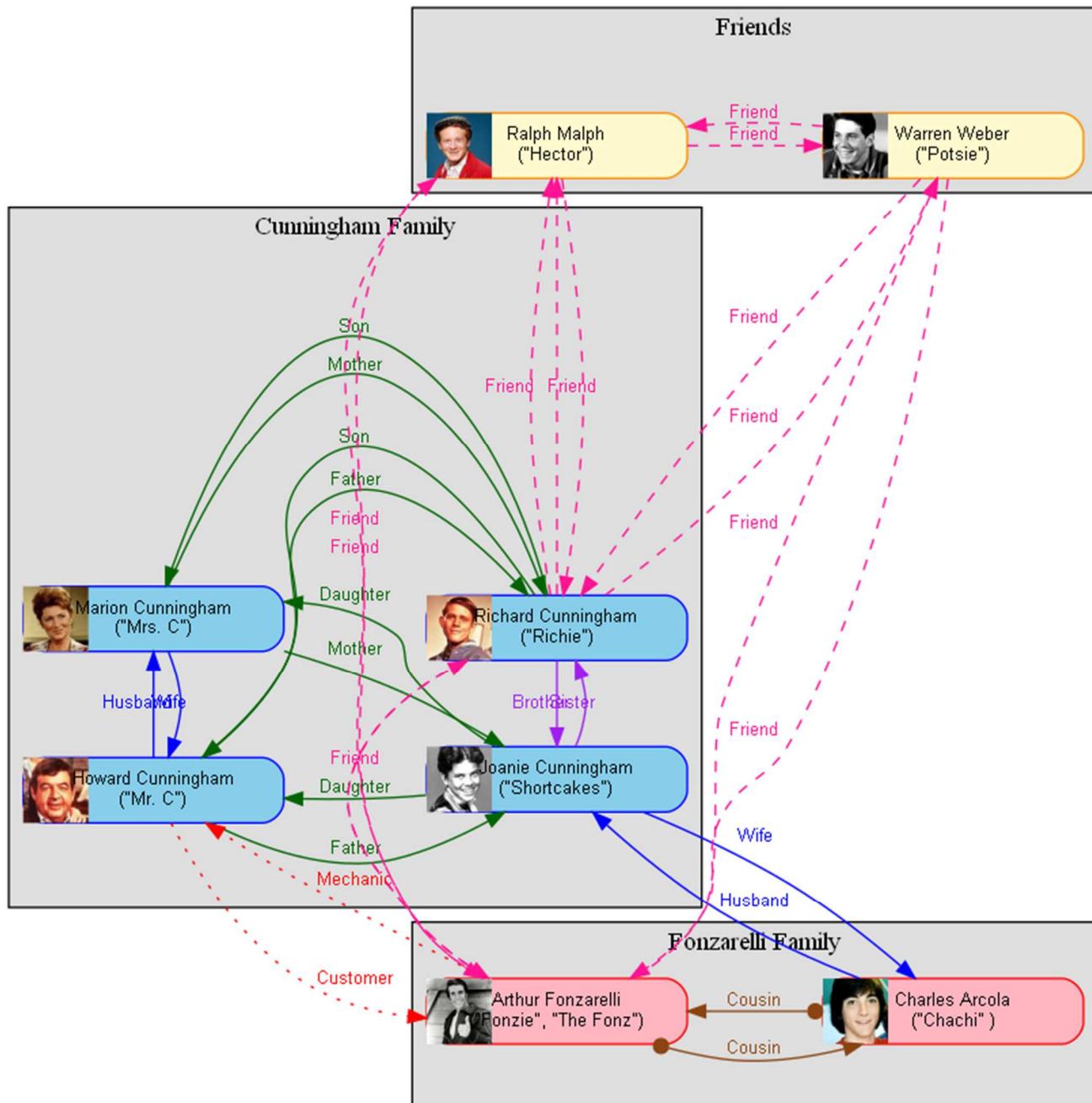


The default layout for dot assumes a top to bottom layout. When the Layout is specified as "dot" a "Direction" dropdown list is dynamically added to the ribbon tab.

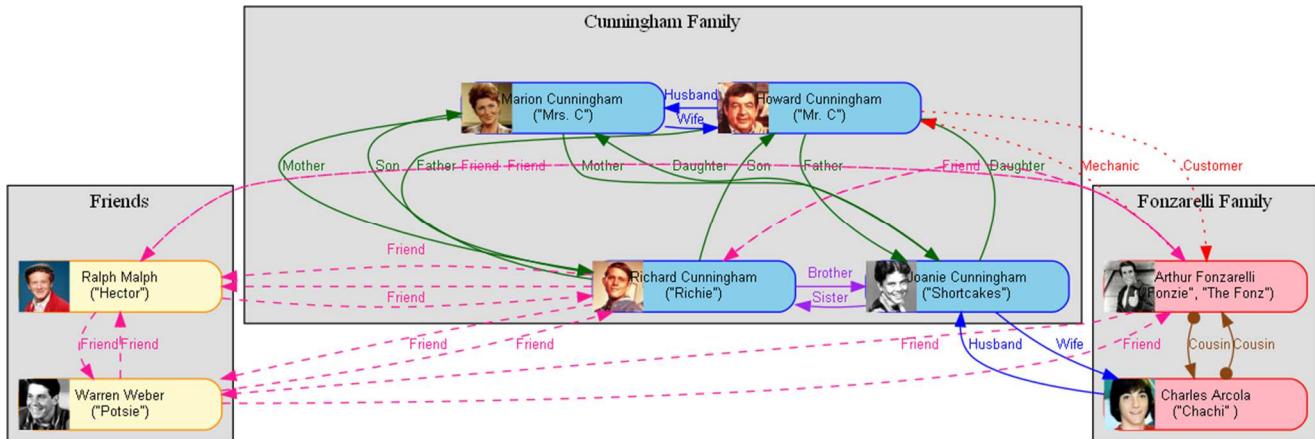




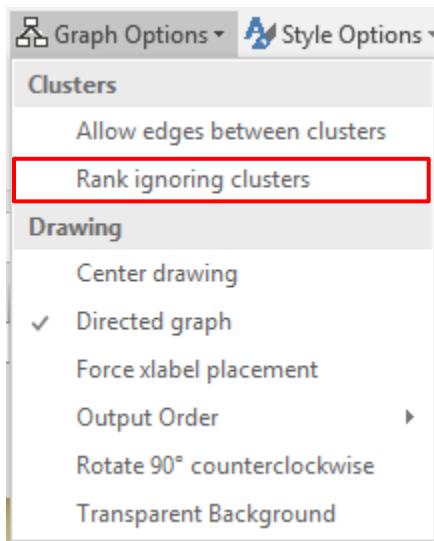
Choose "left to right" from the list of choices and the graph will appear as:



I prefer the previous layout, so select "top to bottom" from the Direction dropdown list. The graph returns to:



Having seen what different layout algorithms can do, let's see what fine tuning some of the graph and edge options can accomplish. Return to the 'Graphviz' tab and select "Rank Ignoring Clusters" from the "Graph Options" dropdown list.

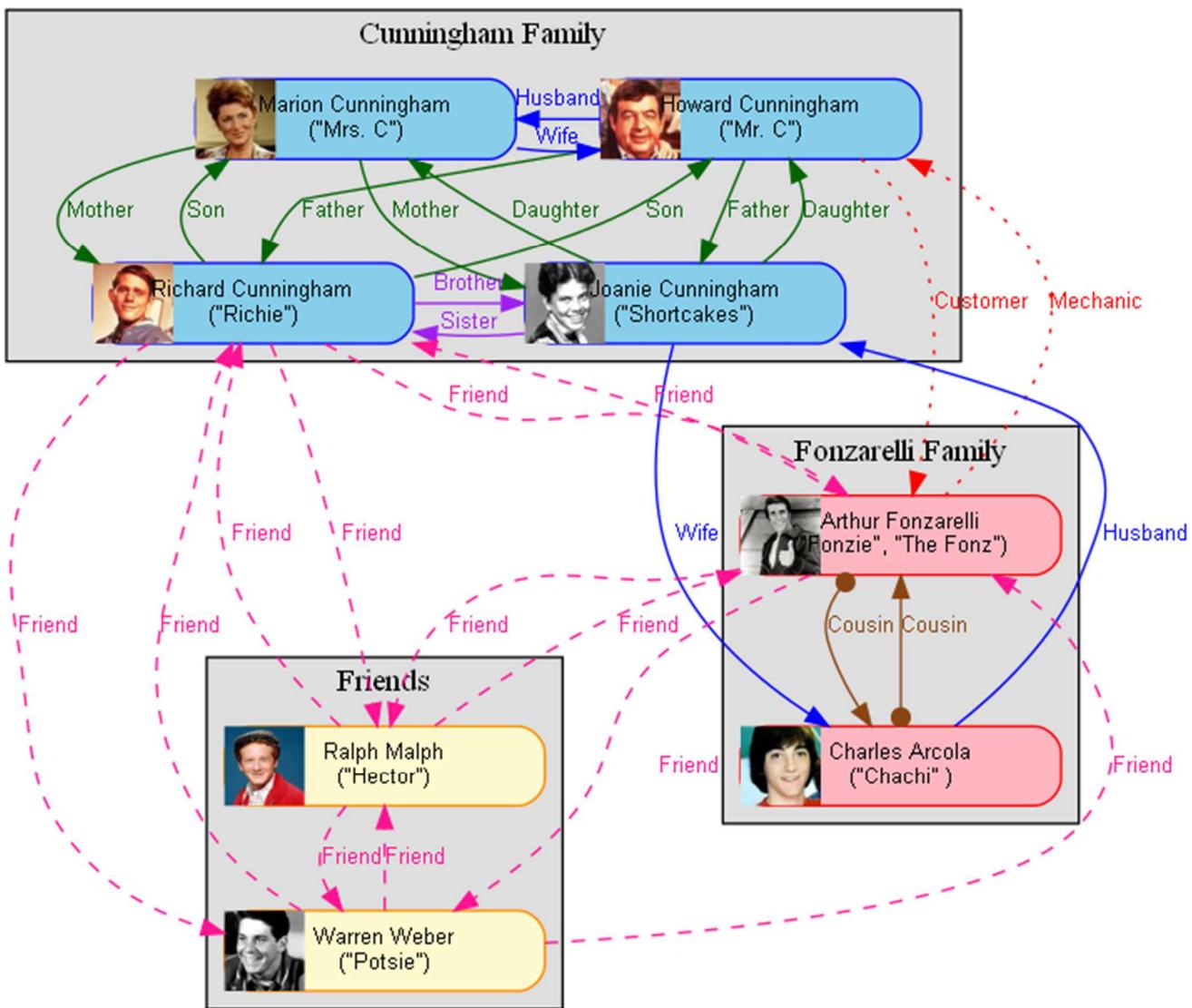


Checking this option causes the graph attribute newrank="true" to be added to the graph options. The newrank attribute specifies whether to use a single global ranking, ignoring clusters.

The original ranking algorithm in dot is recursive on clusters. This can produce fewer ranks and a more compact layout, but sometimes at the cost of a head node being placed on a higher rank than the tail node. It also assumes that a node is not constrained in separate, incompatible subgraphs. For example, a node cannot be in a cluster and also be constrained by rank=same with a node not in the cluster.

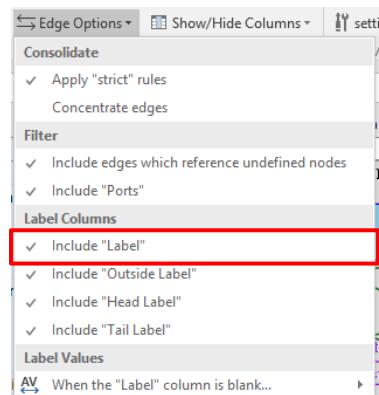
This allows nodes to be subject to multiple constraints. Rank constraints will usually take precedence over edge constraints.

The graph is now more compact, and appears as:

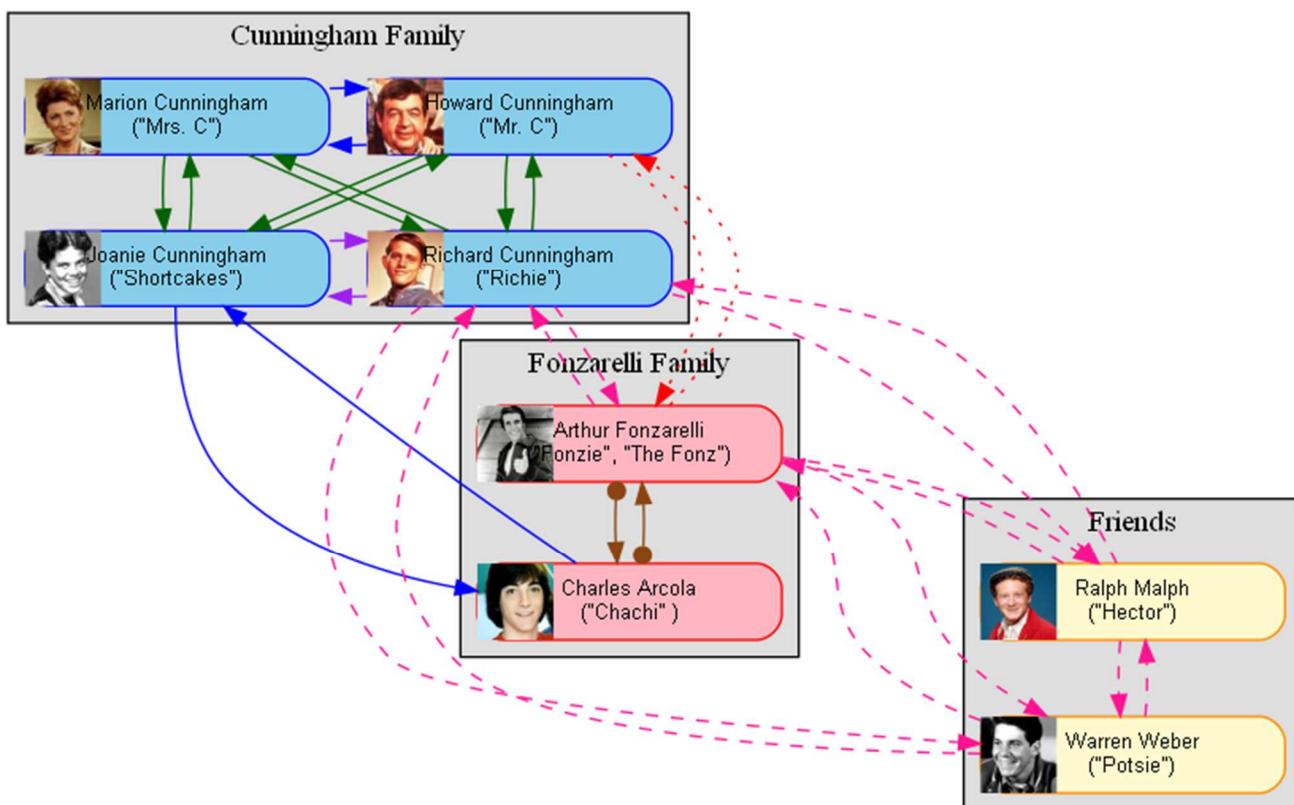


The edge labels have served their purpose, so let's hide them. On the "Edge Options" dropdown list within the Label Columns section you can turn on or off the 4 different types of labels (Label, Outside Label (i.e., xlabel), Head, and Tail) which Graphviz supports.

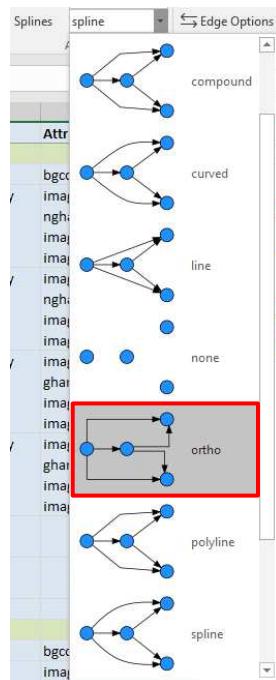
Remove the checkmark from 'Include "Label"' under the 'Label Column' heading



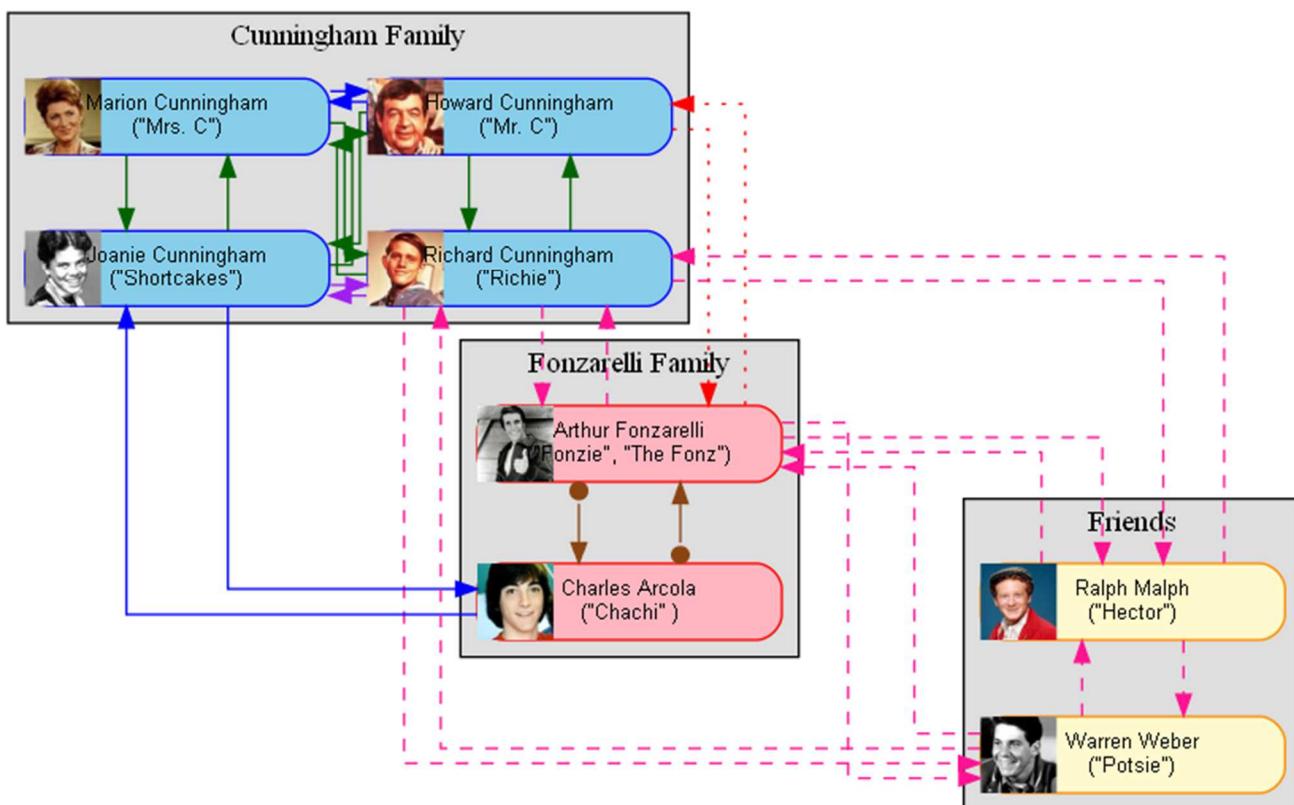
The relationship labels are removed, and the graph now appears as:



Now let's see what effect changing the spline attribute has on the graph. The 'Splines' dropdown list contains examples of how a particular value will cause the edges to be drawn. Choose 'ortho' for orthogonal edges.



The graph now appears as:

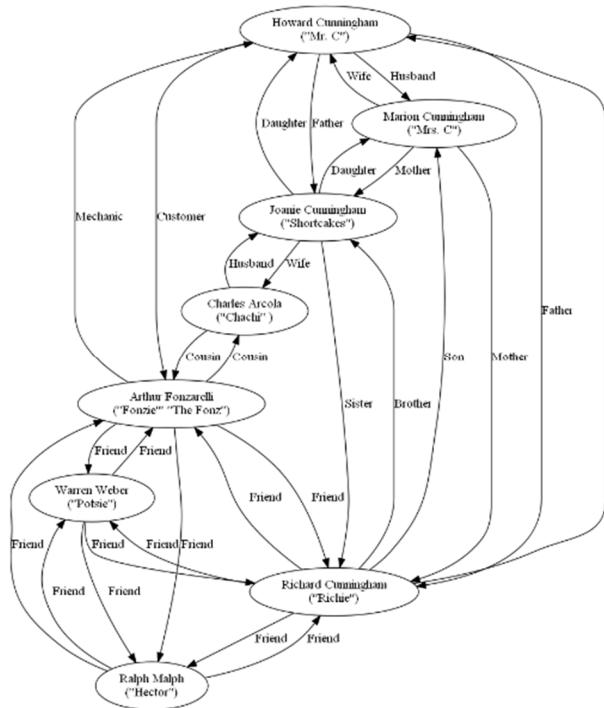


And this representation is a good place to stop.

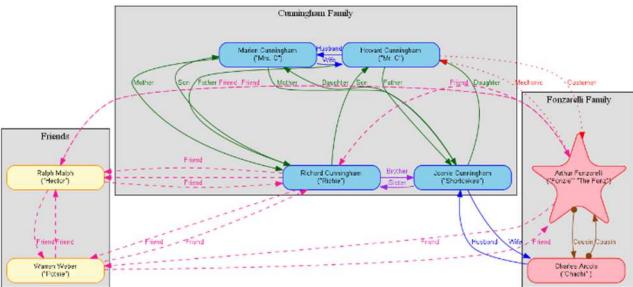
Summary

In summary, we quickly built a relationship graph using Excel column data. Then using the "style designer" worksheet styles were placed in the "styles" worksheet which we associated to rows of data in the "data" worksheet. Additional style attributes were added to individual rows.

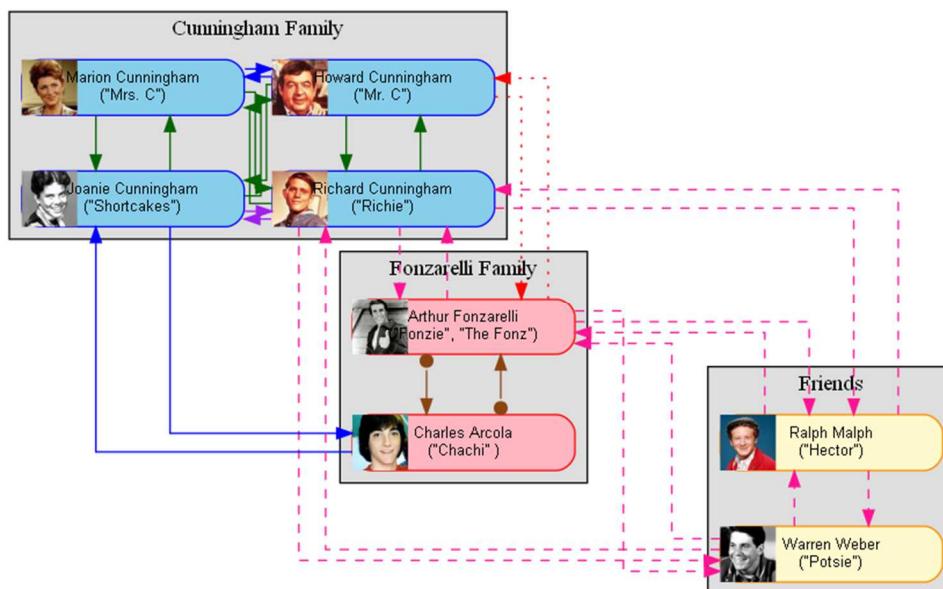
These actions have allowed us to transform the visualization from this default graph:



to this style-based depiction:



and finally, this style-based depiction with row and graph option refinements:



CREATING VIEWS

Introduction

The Relationship Visualizer is capable of handling much larger data sets than can be illustrated in this manual. As the data sets get larger it sometimes becomes useful to see just certain views of the data. The Relationship Visualizer supports views by providing a mechanism on the "styles" worksheet which provides "Yes" or "No" switches to indicate which data to include in the graph.

Views vs. Layers

In this manual, the term "Views" is meant to represent a redrawing of the graph with just a subset of nodes and edges. Graphviz likes to optimize the layout of the nodes and edges based upon the connections, so often the shapes will move to new locations. Contrast this to the term "Layers". Layers are a concept in tools such as Microsoft Visio where shapes can be layered in a manner so that they can be hidden or exposed. In a layered tool, the shapes never change position; they are just visible or invisible. Graphviz also has a concept of layers, but Graphviz layers are outside the scope of this manual.

The Relationship Visualizer spreadsheet's 'styles' worksheet specifies two columns for every style definition. One column has the heading "All Styles" where every row containing a style definition contains the value "Yes", and a column with heading "No Clusters" where any row with style definition of Style Type subgraph-open or subgraph-close contains the value of "No". When the graph is created, any row with a Style having a value of "Yes" is included in the graph, and any row with a Style having a value of "No" is omitted from the graph.

The column used to make these decisions is specified on the Graphviz tab in the dropdown list named "Graph View". The "All Styles" column is chosen by default. Our graph looks like this with "All Styles" selected.

Excel to Graphviz for Win OS

If the Graph View is changed to "No Clusters" the graph is redrawn and appears as

Item	Label	Related Item	Style Name	Attributes
# Cunningham Family				
Howard	Howard Cunningham ("Mr. C")		Cunningham Family	bgcolor="gray87" image="Howard_Cunningham.jpg" imagescale="true" imagepos="ml" style="border: 1px solid black; border-radius: 50%; width: 30px; height: 30px; vertical-align: middle;">
Marion	Marion Cunningham ("Mrs. C")		Cunningham Family	bgcolor="gray87" image="Marion_Cunningham.jpg" imagescale="true" imagepos="ml" style="border: 1px solid black; border-radius: 50%; width: 30px; height: 30px; vertical-align: middle;">
Richie	Richard Cunningham ("Richie")		Cunningham Family	bgcolor="gray87" image="Richard_Cunningham.jpg" imagescale="true" imagepos="ml" style="border: 1px solid black; border-radius: 50%; width: 30px; height: 30px; vertical-align: middle;">
Joanie	Joanie Cunningham ("Shortcake")		Cunningham Family	bgcolor="gray87" image="Joanie_Cunningham.jpg" imagescale="true" imagepos="ml" style="border: 1px solid black; border-radius: 50%; width: 30px; height: 30px; vertical-align: middle;">
>				[rank="same"; "Howard"; "Marion"]
>				[rank="same"; "Richie"; "Joanie"]
}				
# Fonzarelli Family				
Fonzie	Arthur Fonzarelli ("Fonzie", "The Fonz")		Fonzarelli Family	bgcolor="gray87" image="Arthur_Fonzarelli.jpg" imagescale="true" imagepos="ml" style="border: 1px solid black; border-radius: 50%; width: 30px; height: 30px; vertical-align: middle;">
Chachi	Charles Arcola ("Chachi")		Fonzarelli Family	bgcolor="gray87" image="Charles_Arcola.jpg" imagescale="true" imagepos="ml" style="border: 1px solid black; border-radius: 50%; width: 30px; height: 30px; vertical-align: middle;">
)				
# Friends				
Ralph	Ralph Malph ("Hector")		Friends	bgcolor="gray87" image="Ralph_Malph.jpg" imagescale="true" imagepos="ml" style="border: 1px solid black; border-radius: 50%; width: 30px; height: 30px; vertical-align: middle;">

Continuing Our Example with a Custom View

Let us continue our "Happy Days" example. We have built a data set and a graph of the primary characters and their relationships, but what if we want to see a subset graph of just who is married? The way we do that is by defining a column of View switches in the 'styles' worksheet, and direct the Relationship Visualizer to create the graph using just the styles enabled in that column.

Let us create a view for 'Spouse' relationships

Step 1 - Switch to the 'styles' worksheet.

Step 2 - Copy Column E with heading "All" to Column G. The 'styles' worksheet will appear as follows:

A	B	C	D	E	F	G
	Style Name	Format	Style Type	All Styles	No Clusters	All Styles
1						
37	Output	shape="parallelogram" height="0.75" width="1.25" fixedsize="True" fillcolor="White" fontname="Calibri" fontsize="10" style="filled"	node	yes	yes	yes
38	Point	shape="point"	node	yes	yes	yes
39	Predefined Process	shape="rectangle" height="0.75" width="1.25" fixedsize="True" fillcolor="White" fontname="Calibri" fontsize="10" style="diagonals,filled"	node	yes	yes	yes
40	Preparation	shape="hexagon" height="0.75" width="1.25" fixedsize="True" fillcolor="White" fontname="Calibri" fontsize="10" style="filled"	node	yes	yes	yes
41	Process	shape="rect" height="0.75" width="1.25" fixedsize="True" fillcolor="White" fontname="Calibri" fontsize="10" style="filled"	node	yes	yes	yes
42	Small Circle	shape="circle" height="0.25" fixedsize="True" style="filled" penwidth="1" fontname="Arial" fontsize="8"	node	yes	yes	yes
43	Stop	shape="octagon" height="0.75" width="0.75" fixedsize="True" color="Red" fillcolor="white" fontname="Calibri" fontsize="10" style="filled"	node	yes	yes	yes
44	Cunningham Family	shape="rect" height="0.5" width="2" fixedsize="True" fillcolor="SkyBlue" color="Blue" fontname="Arial" fontsize="10" fontcolor="Black" style="rounded,filled"	node	yes	yes	yes
45	Fonzarelli Family	shape="rect" height="0.5" width="2" fixedsize="True" fillcolor="LightPink" color="Red" fontname="Arial" fontsize="10" fontcolor="Black" style="rounded,filled"	node	yes	yes	yes
46	Friends	shape="rect" height="0.5" width="2" fixedsize="True" fillcolor="LemonChiffon" color="DarkOrange" fontname="Arial" fontsize="10" fontcolor="Black" style="rounded,filled"	node	yes	yes	yes
47	Spouse	style="solid" color="Blue" fontname="Arial" fontsize="10" fontcolor="Blue"	edge	yes	yes	yes

Step 3 - Change the heading in cell G1 to "Spouses" and change all the switches in Column F from "yes" to "no" where the "Style Type" is "edge" EXCEPT for the row with the style named "Spouse".

The 'styles' worksheet should now appear as:

	A	B	C	D	E	F	G
	Style Name	Format	Style Type	All Styles	No Clusters	Spouses	
1							
46	Friends	shape="rect" height="0.5" width="2" fixedsize="True" fillcolor="LemonChiffon" color="DarkOrange" fontname="Arial" fontsize="10" fontcolor="Black" style="rounded,filled"	node	yes	yes	yes	
47	Spouse	style="solid" color="Blue" fontname="Arial" fontsize="10" fontcolor="Blue"	edge	yes	yes	yes	
48	Parent Child	style="solid" color="DarkGreen" fontname="Arial" fontsize="10" fontcolor="DarkGreen"	edge	yes	yes	no	
49	Friend Of	style="dashed" color="DeepPink" fontname="Arial" fontsize="10" fontcolor="DeepPink"	edge	yes	yes	no	
50	Business Associate	style="dotted" color="Red" fontname="Arial" fontsize="10" fontcolor="Red"	edge	yes	yes	no	
51	Sibling	style="solid" color="Purple" fontname="Arial" fontsize="10" fontcolor="Purple"	edge	yes	yes	no	
52	Cousin	style="solid" color="Chocolate4" dir="both" arrowtail="dot" fontname="Arial" fontsize="10" fontcolor="Chocolate4"	edge	yes	yes	no	

We will also turn off the ability to add Graphviz directives by changing the "native" style's switch from "yes" to "no". Removing the native directives will cause the rank statements to we discussed earlier to be removed.

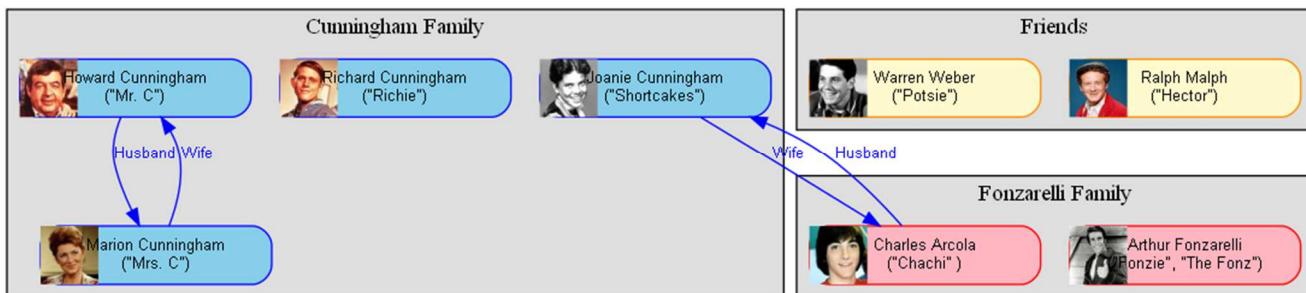
	A	B	C	D	E	F	G
	Style Name	Format	Style Type	All Styles	No Clusters	Spouses	
1							
2	node		node	yes	yes	yes	
3	edge		edge	yes	yes	yes	
4	subgraph-open		subgraph-open	yes	no	yes	
5	subgraph-close		subgraph-close	yes	no	yes	
6	keyword		keyword	yes	yes	yes	
7	native		native	yes	yes	no	

Step 4 - We now have to change a setting on the 'Graphviz' ribbon tab to tell the Relationship Visualizer to use only the styles we have enabled through "yes" settings in column F. Switch to the 'data' worksheet. The 'data' worksheet should appear as follows:

Item	Label	Related Item	Style Name	Attributes
# Cunningham Family				
Howard	Howard Cunningham ("Mr. C")	Cunningham Family	bgcolor="gray7"	image="Howard_Cunningham.jpg"
Marion	Marion Cunningham ("Mrs. C")	Cunningham Family	bgcolor="gray7"	image="Marion_Cunningham.jpg"
Richie	Richard Cunningham ("Richie")	Cunningham Family	bgcolor="gray7"	image="Richie_Cunningham.jpg"
Joanie	Joanie Cunningham ("Shortcakes")	Cunningham Family	bgcolor="gray7"	image="Joanie_Cunningham.jpg"
>	[rank="same"; "Howard"; "Marion"]			imagescale="true"
>	[rank="same"; "Richie"; "Joanie"]			imagepos="ml"
}				
# Fonzarelli Family				
Fonzie	Arthur Fonzarelli ("Fonzie", "The Fonz")	Fonzarelli Family	bgcolor="gray7"	image="Fonzie.jpg"
Chachi	Charles Arcola ("Chachi")	Fonzarelli Family	bgcolor="gray7"	image="Chachi_Arcola.jpg"
}				imagescale="true"
# Friends				imagepos="ml"
Ralph	Ralph Ralph ("Hector")	Friends	bgcolor="gray7"	image="Ralph_Malph.jpg"
			imagescale="true"	imagesize="ml"

Step 5 - Notice that 'Spouses' is now a value in the dropdown list. The dropdown selections in this list are updated every time a new View column is added to the 'styles' worksheet. Change the name of the 'Graph View' from "No Clusters" to "Spouses" in the 'Graphviz' ribbon tab.

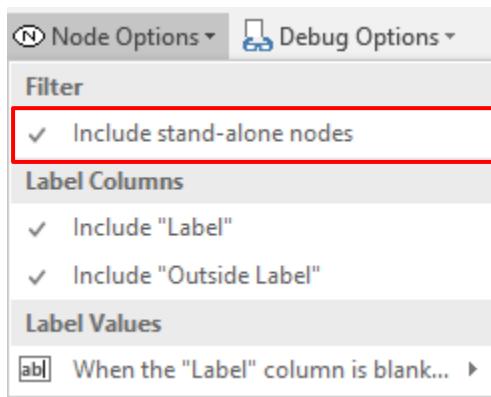
Step 6 - Press the "Refresh Graph" button. The new graph with just the view of the marriage relationships will appear as follows:



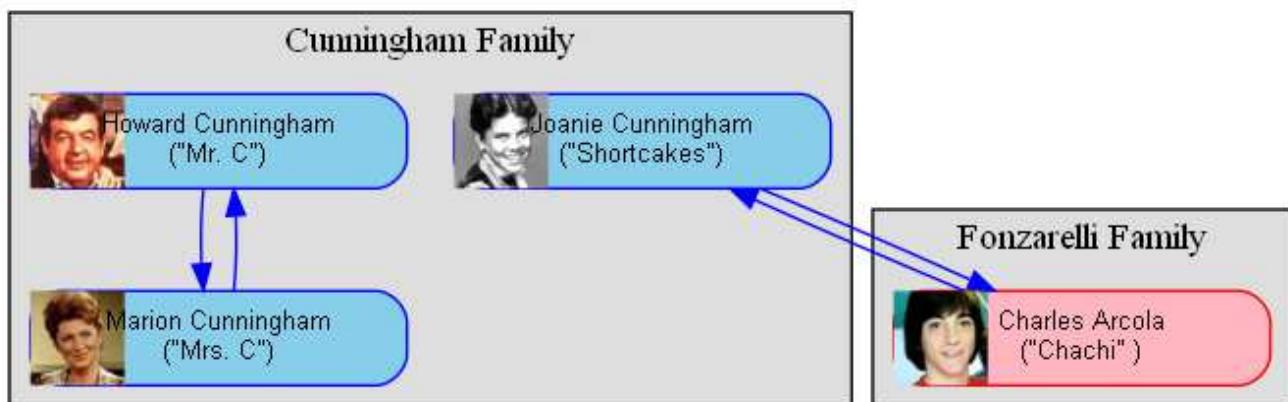
As you can see, the other relationship edges are no longer in the graph. Graphviz has moved the nodes around to make the drawing compact. The graph continues to show all the Happy Days characters; regardless of if they do or do not have a marriage relationship.

Sometimes when the amount of data is large, having nodes in the graph without any edges depicting relationships can be distracting. In the example above, we are interested in the marriage relationships, yet Richard Cunningham, Arthur Fonzarelli, Ralph Malph, and Warren Weber are also depicted. The Relationship Visualizer has switches to remove these island nodes when the island nodes serve no purpose in the graph.

Step 7 - Remove the check mark on the 'Graphviz' ribbon from the 'Nodes' - 'Without Edges' switch control. This means only include nodes that have an edge connection to another node.



Step 8 - Press the "Refresh Graph" button. The graph should now appear as follows:



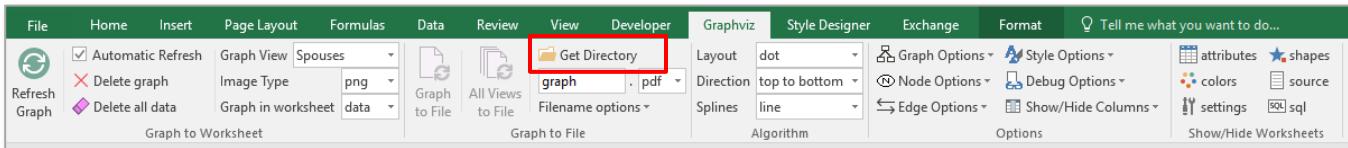
Notice that the "Friends" cluster has disappeared along with the characters without a Spouse relationship. That is because Graphviz does its own filtering and will not draw a cluster that does not contain any nodes.

PUBLISHING GRAPHS

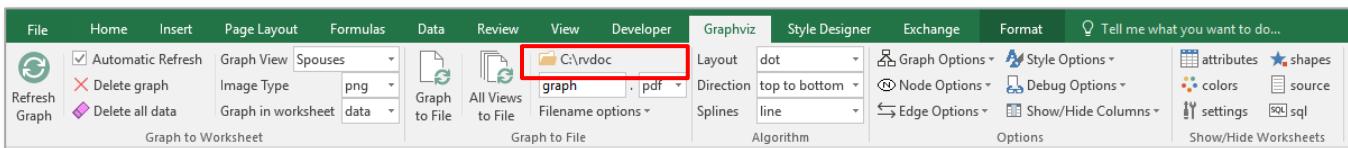
One of the powers of the Relationship Visualizer is that you can collect large amounts of data and let it and Graphviz determine how to graph it efficiently. Graphs with lots of data tend to grow large; much larger than can easily be viewed in Excel. You may also want to print them, and you will need a tool such as Adobe Acrobat Reader that will provide the means to zoom in and out on the graph, and poster print the diagram using multiple sheets of paper.

Setting Output File Options

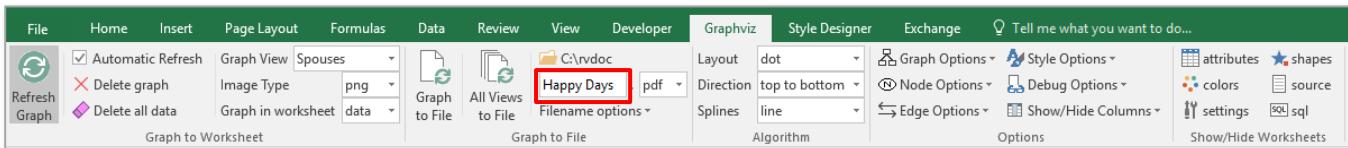
You must specify a directory where you want graph files written to, and provide a filename prefix for the file. Select the 'Get Directory' button on the 'Graphviz' ribbon tab to bring up the Microsoft Windows directory-picking dialog.



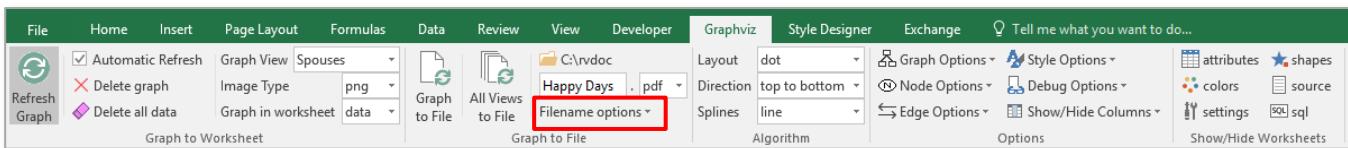
Choose a directory, press OK, and the directory path is transferred to the Output Directory label beneath the 'Get Directory' button. At this point, the 'Graph to File' button is enabled.



Next, specify a File Name Prefix. This is a value that the filename will begin with.



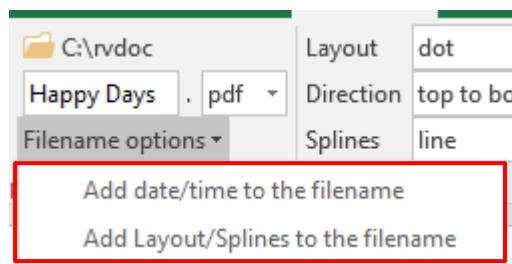
The heading value of the view column used to control which styles were included in the diagram will be appended to the prefix as part of the file name. In this example, we will continue to work with the 'Spouses' view.



Two additional switches can append a date and time to the filename, and the graph options used (layout engine and spline setting) to create the graph. Timestamps on the filename help ensure file creation occurs when you are refining the graphs since Graphviz cannot generate a new file if a file by the same name is

open in another tool, such as Acrobat reader. Recording graph options in the file name allows you to try different layout engines and spline settings top find the combination that produces the best-looking graph.

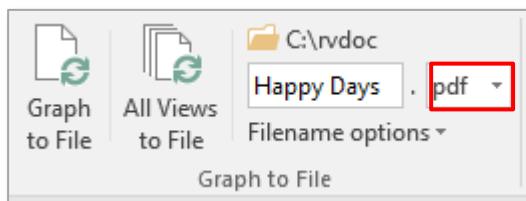
Having this information in the filename is a handy way to recall which settings to use once you have made your decision.



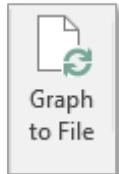
In this example, all the options have been enabled.

Specifying the Output File Format

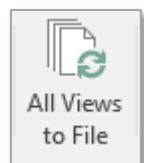
Graphviz provides numerous file formats that the diagrams can be written as, such as **gif**, **jpeg**, **pdf**, or **tiff**. The Relationship Visualizer provides the most commonly used file formats in the 'File Format' dropdown list.



Graph to File / All Views to File



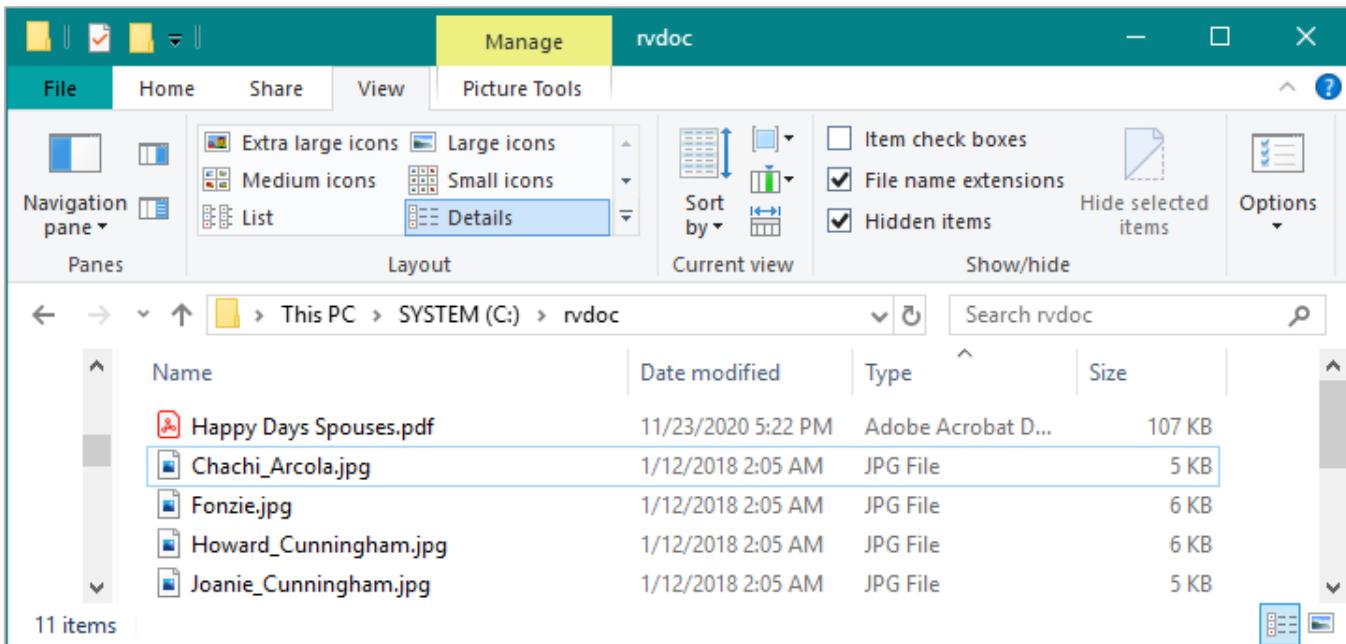
Press the "Graph to File" button. A graph is generated in the same fashion as when "Refresh Graph" was press, but the focus will remain on whatever worksheet was active when the button was pressed (the 'data', 'styles', and 'settings' worksheets all provide this button). You will receive a message down in the status bar that will tell you what the filename is, and where the file is located.



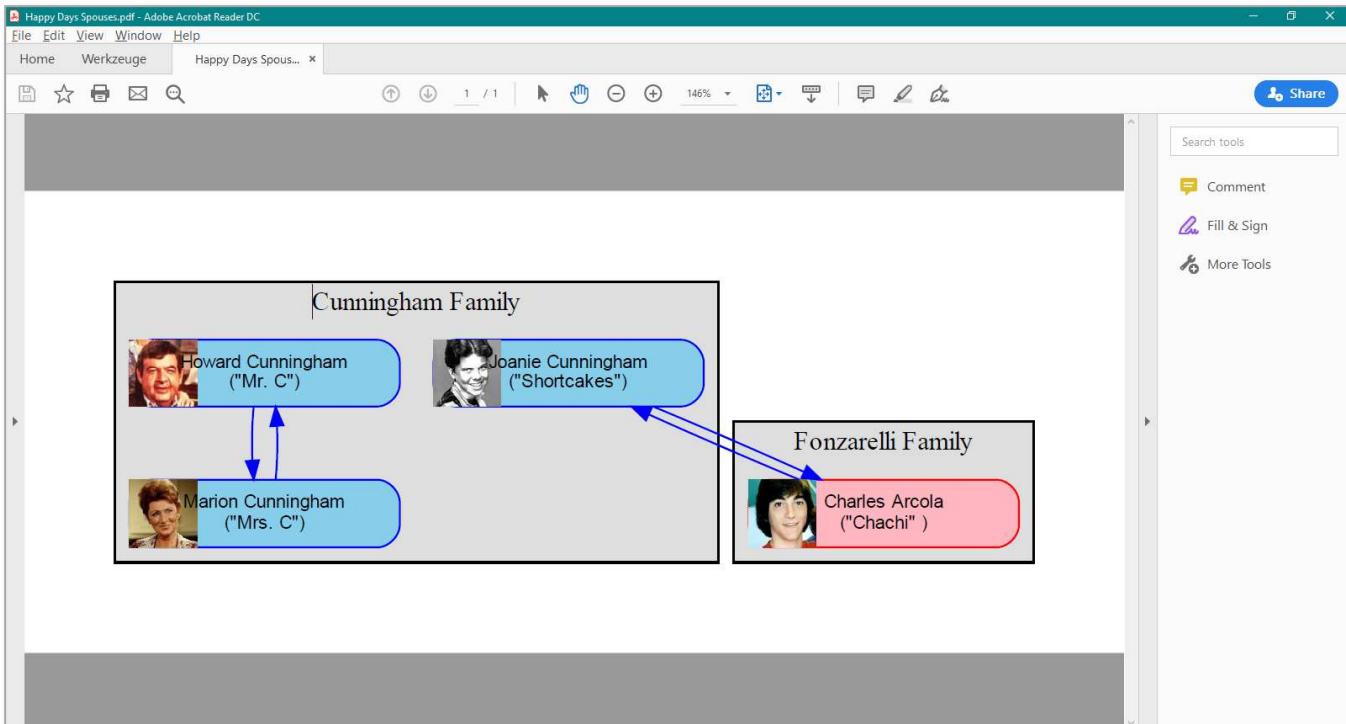
You may also choose to have one file per view created by selecting the "All Views to File" button. The list of views will be iterated in a loop, and a file will be published based on the yes/no switches specified for that view. The View Name will be appended to the file prefix allowing you to tell the graphs apart.

View the File

Bring up Windows Explorer and find your file.



Launch the file, and see it displayed. In this example, the graph is a PDF file. Adobe Acrobat Reader will display the file, and Acrobat Reader's entire zoom, print, and annotation capabilities are available to you.



EXCHANGING DATA

There are several drawbacks of using an Excel workbook with macros as your Graphviz IDE.

- The primary drawback is that the data and the macro code live together. As new versions of the spreadsheet get published with new features it has always been a chore to copy existing data spanning several workbooks and the ribbon settings from the old version to the new version.
- Another drawback is that an Excel Workbook is actually a zip file internally. With the file being a binary file, it is not conducive to checking into version control software such as Git, or for performing a diff on to see what has changed between versions.
- Sharing services such as "pastebin" provide an easy mechanism for sharing content and examples, but they require text files.
- Finally, macro-enabled workbooks are not always trusted by people, or email systems. People trying to share the workbook over email often learn that the email system has stripped the attachment from the email due to the presence of macros.

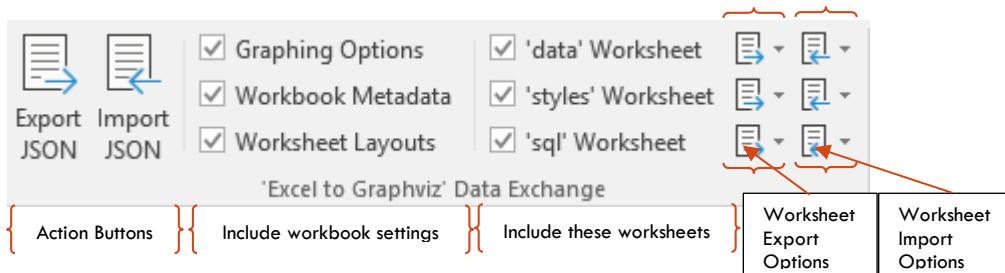
It became apparent that a text-based representation of the data, styles, and settings was needed. The features to support exporting and importing the spreadsheet data comprise the "Exchange" workbook tab. There is not an associated worksheet since data exchange is focused on the internal contents of the workbook.

Let's look at an existing spreadsheet and see how we can export it from one spreadsheet and import it into another using the exchange logic. The "Exchange" tab is the last tab in the workbook, and appears as follows:

Item	Related Item	Style Name
node		Dodecahedron Node
edge		Dodecahedron Edge
A	B,C,D	
B	A,E,G	
C	A,F,I	
D	A,H,J	
E	B,F,K	
F	C,E,M	
G	B,H,L	
H	D,G,O	
I	C,J,N	
J	D,I,P	
K	E,L,Q	
L	G,K,R	
M	F,N,Q	
N	I,M,S	
O	H,P,R	
P	J,O,S	
Q	K,M,T	
R	L,O,T	
S	N,P,T	
T	Q,R,S	

The "Exchange" Ribbon Tab

The "Exchange" tab appears as follows, and is organized as illustrated



- **Export JSON** Writes contents out to JSON file
- **Import JSON** Reads JSON file, and restores data to workbook
- **Graphing Options** Include options chosen in the ribbons and 'settings' worksheet
- **Workbook Metadata** Include information such as user, Excel version, etc.
- **Worksheet Layouts** Include information on how the workbook is organized
- **'data' Worksheet** Include the contents of the 'data' worksheet
- **Export Options**
 - Include Row Number
 - Include Row Height
 - Include Row Visibility
- **Import Options**
 - Append
 - Replace
- **'styles' Worksheet** Include the contents of the 'data' worksheet
- **Export Options**
 - Include Row Number
 - Include Row Height
 - Include Row Visibility
- **Import Options**
 - Append
 - Replace
- **'sql' Worksheet** Include the contents of the 'data' worksheet
- **Export Options**
 - Include Row Number
 - Include Row Height
 - Include Row Visibility
- **Import Options**
 - Append
 - Replace

Exporting Relationship Visualizer Data to JSON format

You can export all the data, or portions of the data depending upon how you intend to use it. Exporting portions of the data is useful when working in teams as you can do things such as export the style definitions and share them, or export the data of team members and combine them into a larger workbook using the 'Append' option of an import.

You have the flexibility to export the entire contents to a JSON file, and selectively import just the sections desired in a new workbook.

Here are example snippets of exported workbook contents:

Graphing Options

<input checked="" type="checkbox"/> Graphing Options	<input checked="" type="checkbox"/> 'data' Worksheet
<input checked="" type="checkbox"/> Workbook Metadata	<input checked="" type="checkbox"/> 'styles' Worksheet
<input checked="" type="checkbox"/> Worksheet Layouts	<input checked="" type="checkbox"/> 'sql' Worksheet

'Excel to Graphviz' Data Exchange

```
{
  "settings": {
    "data": {
      "graphToWorksheet": {
        "runMode": "auto",
        "imageType": "png",
        "imageWorksheet": "data"
      },
      "graphToFile": {
        "directory": "",
        "fileNamePrefix": "graph",
        "imageType": "pdf",
        "appendOptions": false,
        "appendTimeStamp": false
      },
      "layout": {
        "engine": "neato",
        "direction": "top to bottom",
        "rankdir": "TB",
        "splines": ""
      }
    }
  }
}
```

Workbook Metadata

<input checked="" type="checkbox"/> Graphing Options	<input checked="" type="checkbox"/> 'data' Worksheet
<input checked="" type="checkbox"/> Workbook Metadata	<input checked="" type="checkbox"/> 'styles' Worksheet
<input checked="" type="checkbox"/> Worksheet Layouts	<input checked="" type="checkbox"/> 'sql' Worksheet
'Excel to Graphviz' Data Exchange	

```
{  
  "metadata": {  
    "name": "E2GXF",  
    "type": "Excel to Graphviz Exchange Format",  
    "version": "1.0",  
    "user": "Long, Jeffrey",  
    "date": "2020-12-17",  
    "time": "08:43:01",  
    "os": "Windows (32-bit) NT :.00",  
    "excel": "16.0",  
    "filename": "Relationship Visualizer.xlsx"  
  }  
}
```

Worksheet Layouts

- | | |
|---|--|
| <input checked="" type="checkbox"/> Graphing Options | <input checked="" type="checkbox"/> 'data' Worksheet |
| <input checked="" type="checkbox"/> Workbook Metadata | <input checked="" type="checkbox"/> 'styles' Worksheet |
| <input checked="" type="checkbox"/> Worksheet Layouts | <input checked="" type="checkbox"/> 'sql' Worksheet |
- 'Excel to Graphviz' Data Exchange

```
{
  "layouts": {
    "data": {
      "rows": [
        {
          "id": "heading",
          "row": 1,
          "height": 15,
          "hidden": false
        },
        {
          "id": "first",
          "row": 2,
          "height": 15,
          "hidden": false
        }
      ],
      "columns": [
        {
          "id": "flag",
          "column": 1,
          "heading": "",
          "width": 1.71,
          "hidden": false,
          "wrapText": false
        },
        {
          "id": "item",
          "column": 2,
          "heading": "Item",
          "width": 20.71,
          "hidden": false,
          "wrapText": false
        },
        {
          "id": "tailLabel",
          "column": 3,
          "heading": "Tail Label",
          "width": 0,
          "hidden": true,
          "wrapText": true
        },
        {
          "id": "label",
          "column": 4,
          "heading": "Label",
          "width": 0,
          "hidden": true,
          "wrapText": true
        }
      ]
    }
  }
}
```

'data' Worksheet

<input checked="" type="checkbox"/> Graphing Options	<input checked="" type="checkbox"/> 'data' Worksheet
<input checked="" type="checkbox"/> Workbook Metadata	<input checked="" type="checkbox"/> 'styles' Worksheet
<input checked="" type="checkbox"/> Worksheet Layouts	<input checked="" type="checkbox"/> 'sql' Worksheet

'Excel to Graphviz' Data Exchange

```
{  
  "content": {  
    "data": [  
      {  
        "row": 2,  
        "hidden": false,  
        "height": 15  
      },  
      {  
        "row": 3,  
        "hidden": false,  
        "height": 15,  
        "item": "node",  
        "styleName": "Dodecahedron Node"  
      },  
      {  
        "row": 4,  
        "hidden": false,  
        "height": 15,  
        "item": "edge",  
        "styleName": "Dodecahedron Edge"  
      },  
      {  
        "row": 5,  
        "hidden": false,  
        "height": 15,  
        "item": "A",  
        "relatedItem": "B,C,D"  
      }  
    ]  
  }  
}
```

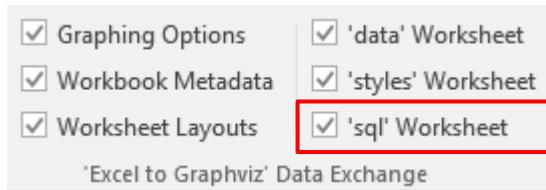
'styles' Worksheet

<input checked="" type="checkbox"/> Graphing Options	<input checked="" type="checkbox"/> 'data' Worksheet
<input checked="" type="checkbox"/> Workbook Metadata	<input checked="" type="checkbox"/> 'styles' Worksheet
<input checked="" type="checkbox"/> Worksheet Layouts	<input checked="" type="checkbox"/> 'sql' Worksheet

'Excel to Graphviz' Data Exchange

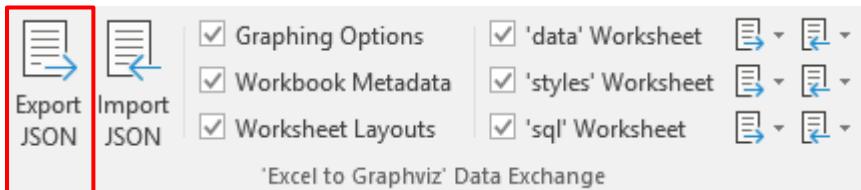
```
{
  "content": {
    "styles": [
      {
        "name": "Dodecahedron Node",
        "type": "node",
        "format": {
          "shape": "circle",
          "width": "0.25",
          "fixedsize": "true",
          "fontname": "Calibri Bold",
          "fontsize": "10",
          "style": "filled",
          "fontcolor": "white",
          "fillcolor": "red"
        },
        "viewSwitches": [
          "yes",
          "yes"
        ]
      },
      {
        "name": "Dodecahedron Edge",
        "type": "edge",
        "format": {
          "fontname": "Calibri",
          "fontsize": "10"
        },
        "viewSwitches": [
          "yes",
          "yes"
        ]
      }
    ]
  }
}
```

'sql' Worksheet

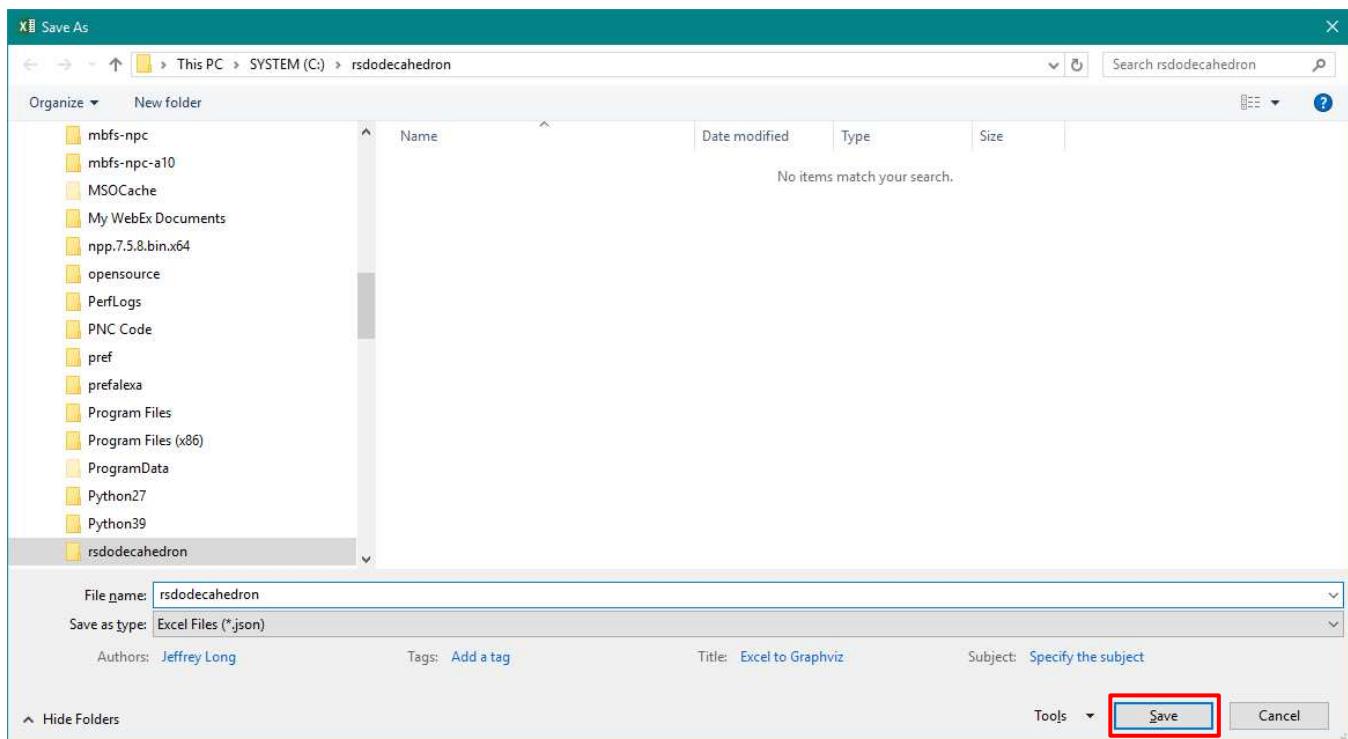


```
{
  "content": {
    "sql": [
      {
        "row": 3,
        "hidden": false,
        "height": 27,
        "sqlStatement": "SELECT 'node' as [ITEM],\n          'Small Circle' as [Style Name],\n          'excelFile': \"usa states.xlsx\",\n          \"status\": \"SUCCESS\""
      },
      {
        "row": 5,
        "hidden": false,
        "height": 40.5,
        "sqlStatement": "SELECT '{' as [ITEM],\n          'USA' as [Label],\n          'Border 1 Begin' as [Style Name],\n          \"excelFile\": \"usa states.xlsx\",\n          \"status\": \"SUCCESS\""
      }
    ]
  }
}
```

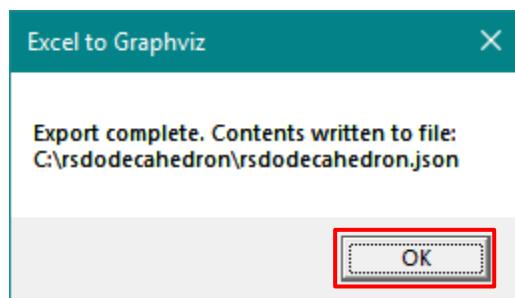
Make the selections of the data you wish to export, and press the "Export JSON" button



You will be prompted to specify the name of a JSON file that the data should be written to. Enter a file name and press the "Save" button.



Once the data is written to the file you will receive a pop-up message such as:



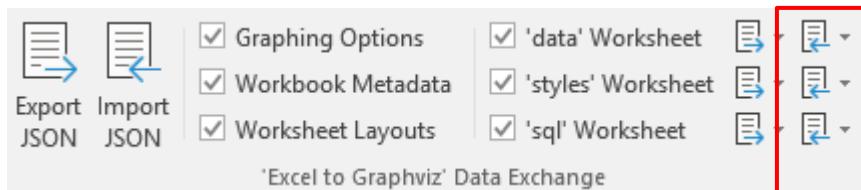
Press the OK button, and you are done.

Importing JSON Data into the Relationship Visualizer

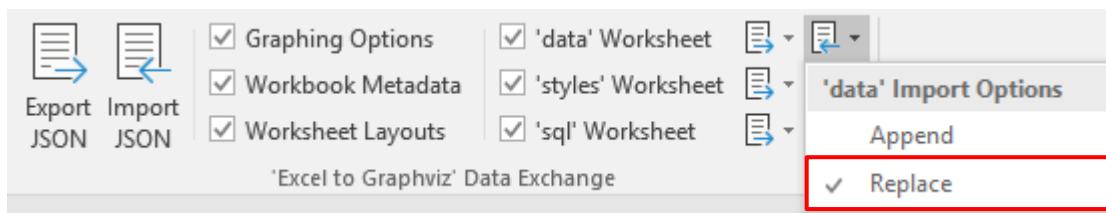
The Import function works the opposite of the Export function. It reads a JSON file which has been exported by the Relationship Visualizer to reconstitute the JSON data into a Relationship Visualizer spreadsheet.

To import a JSON file, start by choosing the sections which you want included. Just as you can export an entire workbook, or sections of the workbook, you may also import an entire workbook or just sections of a workbook.

A key difference for importing worksheets comes via the import option dropdown lists.



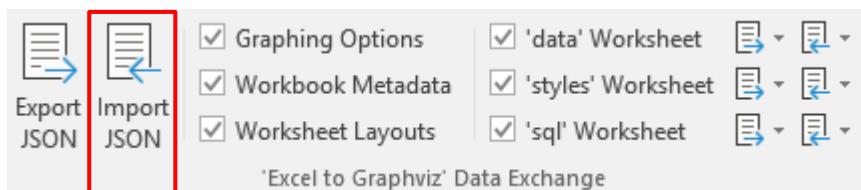
'data', 'styles', and 'sql' worksheets choices have Import options of Append and Replace.



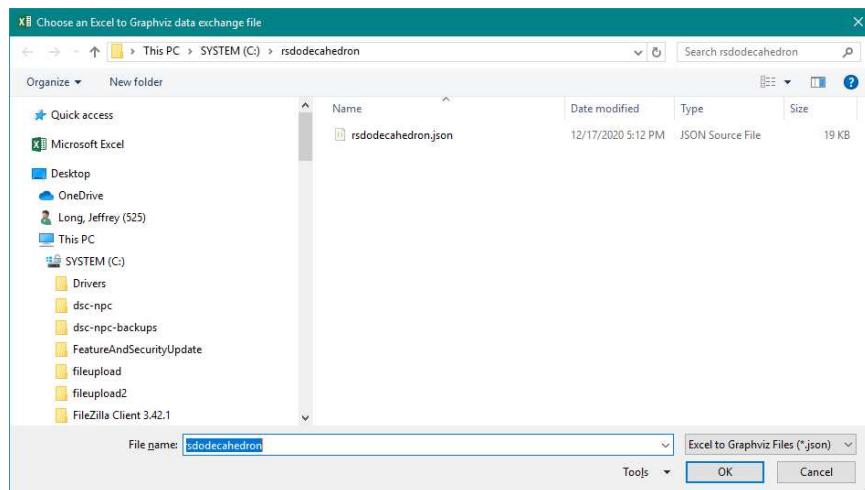
These are mutually exclusive choices that allow you to specify whether to replace the contents in the worksheet, or append the data in the worksheet. **Replace** is the default, and is intended for loading the data into an empty worksheet. **Append** is useful for consolidating data when multiple people are preparing the data.

For example, assume a husband and wife each prepare a family tree of their ancestors. The husband can export his ancestor's data, and the wife can import the husband's data with Append checked. The import will place the data in the 'data' worksheet after the wife's data and the family tree will become complete for this family unit.

Once you have selected your import options, press the "Import JSON" button.



You will be prompted to "Choose an Excel to Graphviz data exchange file"



Select the file and press the OK button. The data will be imported (which may take several seconds). If the "Automatic Refresh" checkbox on the Graphviz tab is checked, the Relationship Visualizer will graph the data to the worksheet, otherwise press the "Refresh Graph" button to see the graph.

Item	Related Item
node	
edge	B,C,D A,E,G C,A,F,I D,A,H,J E,B,F,K F,C,E,M G,B,H,L H,D,G,O I,C,J,N J,D,I,P K,E,I,Q L,G,K,R M,F,N,Q N,I,M,S O,H,P,R P,J,O,S Q,K,M,T R,L,O,T S,N,P,T T,Q,R,S

ADVANCED TOPICS, TIPS, AND TRICKS

This section describes several miscellaneous features in the Relationship Visualizer that can be used to create more elaborate graphs.

HTML Labels

Graphviz has a feature where if the value of a label attribute for nodes, edges, clusters, or graphs is given as an HTML string that is delimited by < . . . >, the label is interpreted as an HTML description. At their simplest, such labels can describe multiple lines of variously aligned text as provided by ordinary string labels. More generally, the label can specify a table like those provided by HTML, with different graphical attributes at each level.

The features and syntax supported by these labels are modeled on HTML. However, there are many aspects that are relevant to Graphviz labels that are not in HTML and, conversely, HTML allows various constructs which are meaningless in Graphviz. The Graphviz creators generally refer to these labels as "HTML labels" but the reader is warned that these labels are not HTML. The grammar which Graphviz will accept is fully described at: <http://www.graphviz.org/doc/info/shapes.html#html>

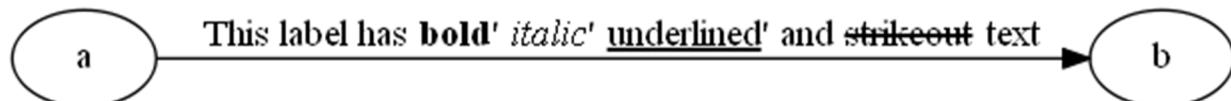
A basic HTML label can be constructed as text between and elements, then wrapped in the < and > delimiters as described above. For example, a label can be constructed as:

```
<<font>This label has <b>bold</b>, <i>italic</i>,
<u>underlined</u>, and <s>strikeout</s>
text</font>>
```

and entered as a label value for an edge. In this example, we will relate 'a' to 'b' as we are interested in seeing how the edge is drawn. The 'data' worksheet appears as:

A	B	C	G
1	Item	Label	Is Related To Item
2	a	<This label has bold, <i>italic</i>, <u>underlined</u>, and <s>strikeout</s> text>	b
3			

Pressing "Refresh Graph" produces the following graph:



A slightly more complex example is to create a HTML table. In this example, the table contains one row with two cells:

```
<<table>
<tr>
  <td>Cell 1</td>
  <td>Cell 2</td>
</tr>
</table>>
```

Using it to represent a node named 'c', the 'data' worksheet appears as:

	A	B	C	G
1	Item	Label	Is Related To Item	
2	a	<<table> <tr> <td>Cell 1</td> <td>Cell 2</td> </tr> </table>>		
3				

Pressing "Refresh Graph" produces the following graph:

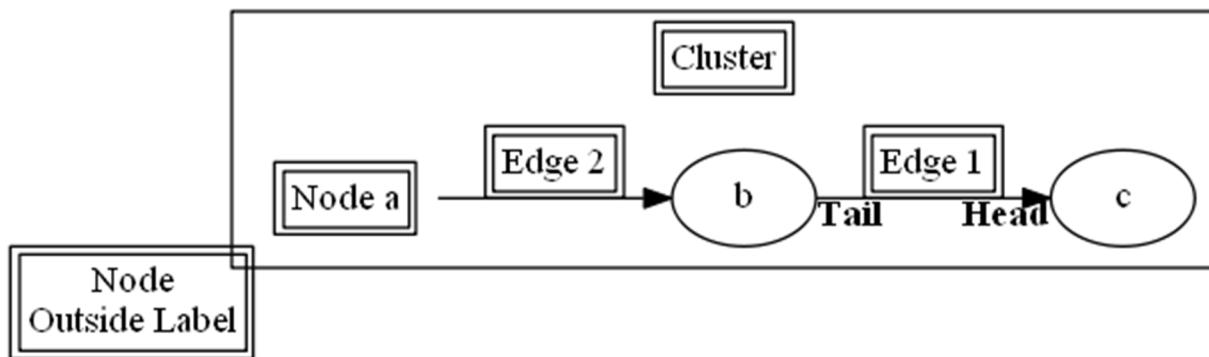


HTML labels can be used for Clusters, Nodes, and Edges. In the example below there are three Items named 'a', 'b', and 'c'. HTML labels have been added for node 'a', and the edges from 'a' to 'b' and from 'b' to 'c'. The nodes and edges are wrapped with a border via a cluster that also has an HTML label.

The 'data' worksheet appears as:

	A	B	C	D	E	F	G
1	Item	Label	Outside Label	Tail Label	Head Label	Is Related To Item	
2	{	<<table> <tr><td>Cluster</td></tr></table>>					
3	a	<<table> <tr><td>Node a</td></tr></table>>	<<table> <tr><td>Node Outside Label</td></tr></table>>				
4	a	<<table> <tr><td>Edge 2</td></tr></table>>				b	
5	b	<<table> <tr><td>Edge 1</td></tr></table>>		<Tail> <Head>	c		

Pressing "Refresh Graph" produces the following graph:



Keywords - 'Graph', 'Node', and 'Edge' Keywords

Graphviz has a built-in behavior where if a default attribute is defined using a node, edge, or graph statement, any object of the appropriate type defined afterwards will inherit this attribute value. This holds until the default attribute is set to a new value, from which point the new value is used. Objects defined before a default attribute is set will have an empty string value attached to the attribute once the default attribute definition is made.

The Relationship Visualizer also supports this capability by reserving the values "graph", "node", and "edge" as keywords in the 'Item' column of the 'data' worksheet. Appropriate syntax statements are added to the DOT source code to put the styles defined by the values in the "Style" and "Attributes" columns into effect (when these columns are enabled on the 'settings' worksheet) when these keywords are detected.

In the following example, nodes have been defined with an Item ID of 'a' through 'h'. On row 5 a node statement has been placed with an "Attributes" definition which changes the font color to red (the node statement has conditional formatting which changes the cell background color and changes the font to bold italic to differentiate the keyword from ordinary data). All nodes from that point forward are rendered with a red font. This continues until a second node statement is encountered on row 10 that resets the font color to a null string, which tells Graphviz to resume using the default value.

A	B	D	G	H	I
1	Item	Label	Related Item	Style Name	Attributes
2					
3	a	ant			
4	b	bat			
5	node				fontcolor="red"
6	c	cat			
7	d	dog			
8	e	eel			
9	f	frog			
10	node				fontcolor=""
11	g	giraffe			
12	h	hippo			
13					

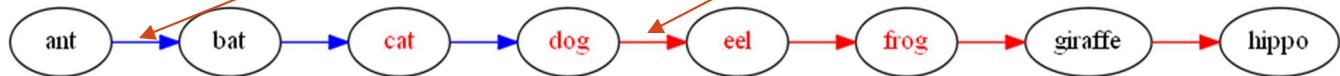
Pressing "Refresh Graph" produces the following graph:



Likewise, this same capability exists for edges using the "edge" keyword. In the example below an edge keyword on row 13 sets the edge color to blue for the first 3 edges. A second edge keyword on row 17 changes the color to red for all the remaining edges.

A	B	D	G	H	I
1	Item	Label	Related Item	Style Name	Attributes
2					
3	a	ant			
4	b	bat			
5	node				fontcolor="red"
6	c	cat			
7	d	dog			
8	e	eel			
9	f	frog			
10	node				fontcolor=""
11	g	giraffe			
12	h	hippo			
13	edge				color="blue"
14	a		b		
15	b		c		
16	c		d		
17	edge				color="red"
18	d		e		
19	e		f		
20	f		g		
21	g		h		

Produces the following graph:



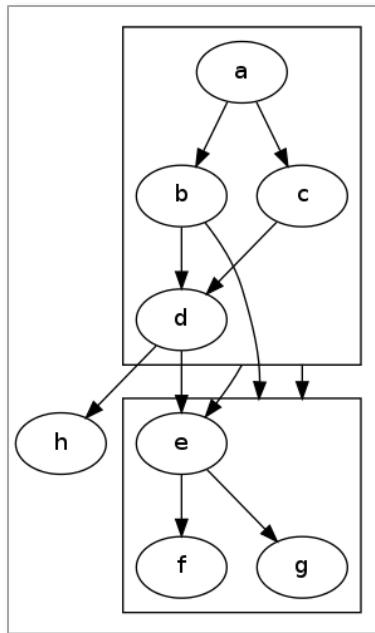
Note that a subgraph receives the attribute settings of its parent graph at the time of its definition. This can be useful; for example, one can assign a font to the root graph and all subgraphs will also use the font. For some attributes, however, this property is undesirable. If one attaches a label to the root graph, it is probably not the desired effect to have the label used by all subgraphs. Rather than listing the graph attribute at the top of the graph, and the resetting the attribute as needed in the subgraphs, one can simply defer the attribute definition in the graph until the appropriate subgraphs have been defined.

Clusters - Depicting a Relationship from or to a Cluster

You may have a situation where you are trying to represent a dependency diagram where you have nodes inside a cluster, and you want to be able to make nodes and/or clusters dependent on other nodes and/or clusters. In other words, you want an edge to begin and/or end at the border of a cluster. This goal can be accomplished in the Relationship Visualizer with a little bit of additional Graphviz knowledge.

Let's reproduce the diagram below which can be found at:

<http://stackoverflow.com/questions/2012036/graphviz-how-to-connect-subgraphs>



Let's start by putting the 'graph', 'node' and 'edge' keyword features described in the previous section into a tangible example. In rows 4-6 below each keyword is specified along with style formatting for each object. These statements are not required to connect clusters, but are illustrative of the keyword feature which was just described, and will help visually distinguish elements of the graph. For graphs, nodes, and edges set the "Attributes" cells with the following formatting information:

- **Graph:** fontname="Arial" fontsize="12" fontcolor="red"
- **Node:** fontname="Arial"
- **Edge:** fontname="Arial" fontsize="8" decorate="true" color="blue"

Excel to Graphviz for Win OS

The spreadsheet should look as follows:

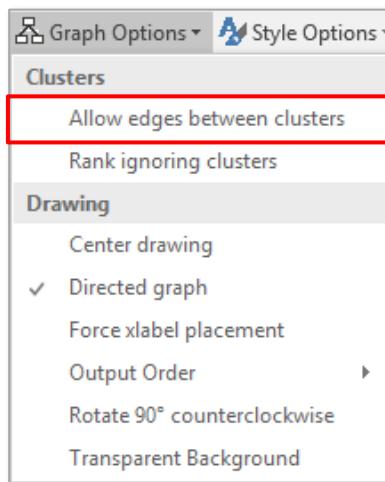
A	B	C	G	H	I
1	Item	Label	Is Related To Item	Style Name	Extra Style Attributes
2	# Set the font and font size for clusters, nodes, and edges				
3	graph				fontname="Arial" fontsize="12" fontcolor="Red"
4	node				fontname="Arial"
5	edge				fontname="Arial" fontsize="8" color="Blue" decorate="true"

Next we need we enable the "compound" graph attribute which Graphviz sets to 'false' by default. Setting it to 'true' activates a Graphviz feature allowing edges to connect to clusters. The easiest way to do this is to follow the instructions previously described in **Adding Native Graphviz Directives** on page 127 to add a native Graphviz statement. Place a '>' character in the Item column to indicate this is a native command and enter compound="true" in the Label column as shown below.

6	# Enable 'compound' graph attribute			
7	>	compound="true"		

The compound="true" statement is then added to the body of the main graph.

You can also achieve the same result by checking the graph option "Allow edges between clusters"



Next, define 2 clusters. To make the example clear we will label them as cluster0 and cluster1. Cluster0 will have 4 edge relationships using the letters a, b, c, and d as node names which will fall inside the cluster border. Likewise, cluster1 will have edge relationships using the letters e, f, and g as node names that fall inside a cluster border.

Up until this point we have always defined the start of a cluster with an open brace '{'. When the Relationship Visualizer sees an open brace, it generates an internal name for the cluster to makes things simpler for the user. A cluster must have an Item name to reference it by if we are to connect edges to it, so using the macro-

generated name could be haphazard, as the name will change if the cluster definition moves around within the Excel spreadsheet.

The Relationship Visualizer solves that problem by allowing you to provide a name preceding the open brace (for example, "cluster0") to designate a named subgraph. If the name of the subgraph begins with **cluster**, Graphviz notes the subgraph as a special cluster subgraph. If supported, the layout engine will do the layout so that the nodes belonging to the cluster are drawn together, with the entire drawing of the cluster contained within a bounding rectangle. Note that, for good and bad, cluster subgraphs are not part of the DOT language, but solely a syntactic convention adhered to by certain of the layout engines. If the name does not begin with **cluster**, then Graphviz treats the subgraph as an ordinary subgraph.

The spreadsheet now looks as follows to define the clusters. Note in the illustration that the occurrence of the cluster name in the Item and Label columns. We will remove the label later in this example, but for now it helps make the lesson clearer.

8	# cluster0			
9	cluster0 {	cluster0		
10	a		b	
11	a		c	
12	b		d	
13	c		d	
14	}			
15	# cluster1			
16	cluster1 {	cluster1		
17	e		g	
18	e		f	
19	}			

Once the clusters have been defined, we can specify edges that show relationships between the clusters. Five scenarios are depicted:

1. A relationship from a node within a cluster to another node within a cluster (row 21)
2. A relationship from a node within a cluster to a node outside of a cluster (row 22)
3. A relationship from a node within a cluster to the border of another cluster (row 23)
4. A relationship from the border of a cluster to a node within a cluster (row 24)
5. A relationship from the border of a cluster to the border of another cluster (row 25)

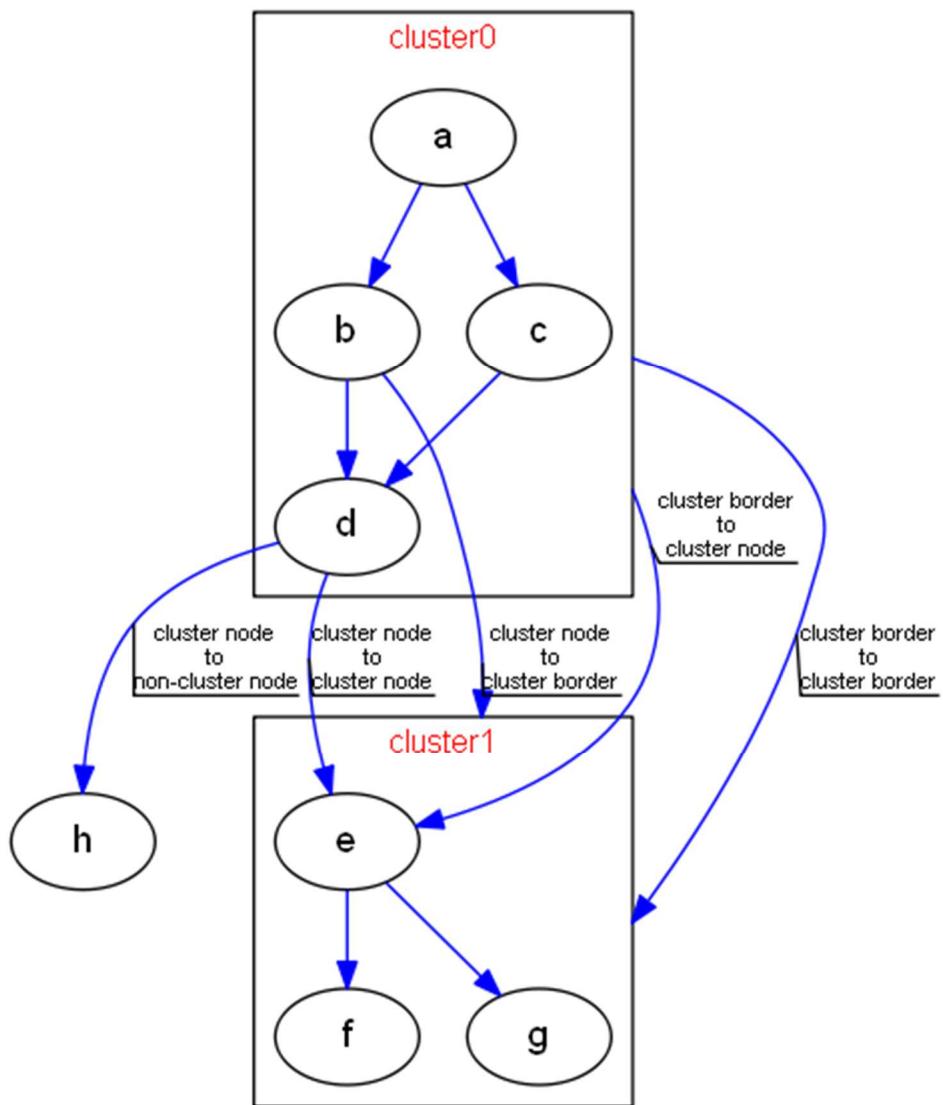
To specify the cluster where the edge should originate from you must specify a **ltail** attribute in the Attributes column which specifies the name of the cluster the edge should originate from. For example, **ltail="cluster0"**. Note that the item name in the "Item Name" column must reside within that cluster.

When you want the arrowhead of an edge to stop at the border you must specify a **lhead** attribute in the Attributes column which specifies the name of the cluster to connect to. For example, **lhead="cluster1"**. Note that the item name in the "Related Item" column must reside within that cluster.

The spreadsheet appears as follows to show the five scenarios described above. Descriptions of the scenarios have been added in the Label column to help explain this example. Later in the example we will hide them.

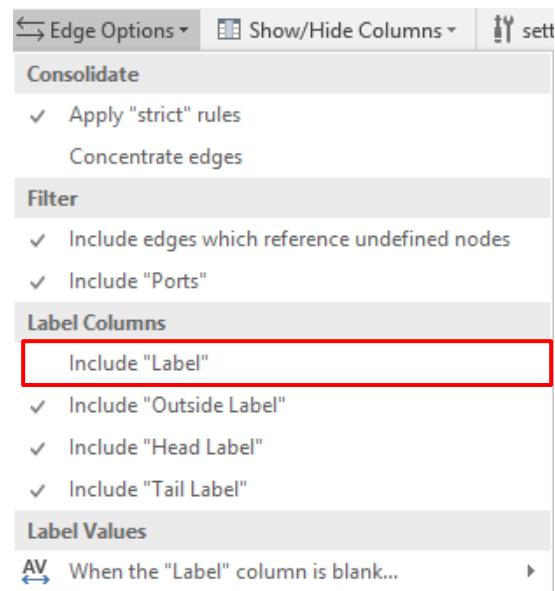
20	#	Edge scenarios			
21	d	cluster node to cluster node	e		
22	d	cluster node to non-cluster node	h		
23	b	cluster node to cluster border	f		!head="cluster1"
24	c	cluster border to cluster node	e		!tail="cluster0"
25	c	cluster border to cluster border	g		!tail="cluster0" !head="cluster1"

At this point all the information necessary to create the graph has been entered. When you press the "Refresh Graph" button the graph appears as follows:

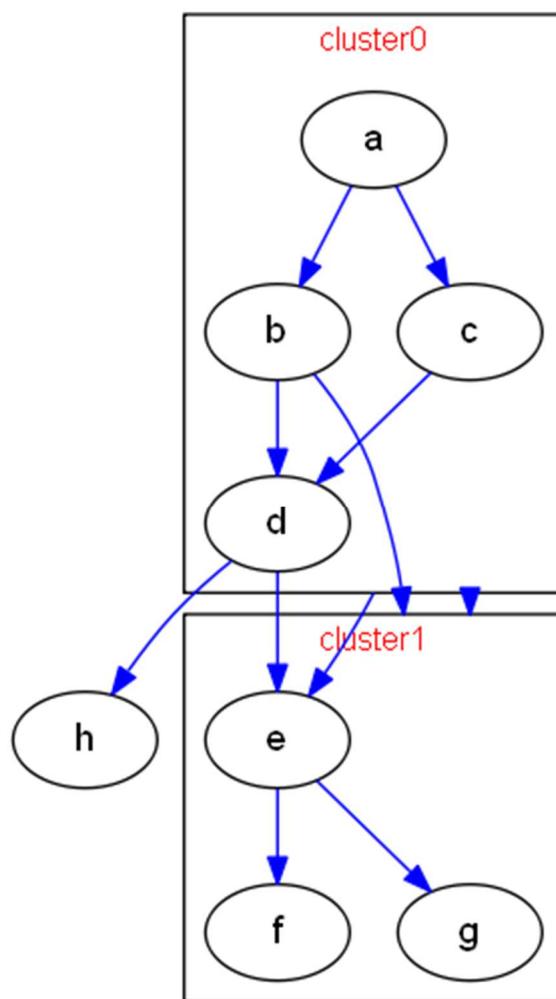


The graph looks like our desired graph. Notice that each blue edge has a black line attached which underlines the label text, and the edge label describes the relationship depicted. The callout line effect was achieved by us adding the attribute `decorate="true"` when the edge keyword was defined at the top of the spreadsheet. It was useful here to help see the scenario depicted as Graphviz sometimes places labels in locations which can cause confusion.

Now that we see how the edge statements are graphed, we can hide the labels. An easy way to do that is to go to the 'Graphviz ribbon tab, and in the "Edge Options" remove the check box from **Include Label**".



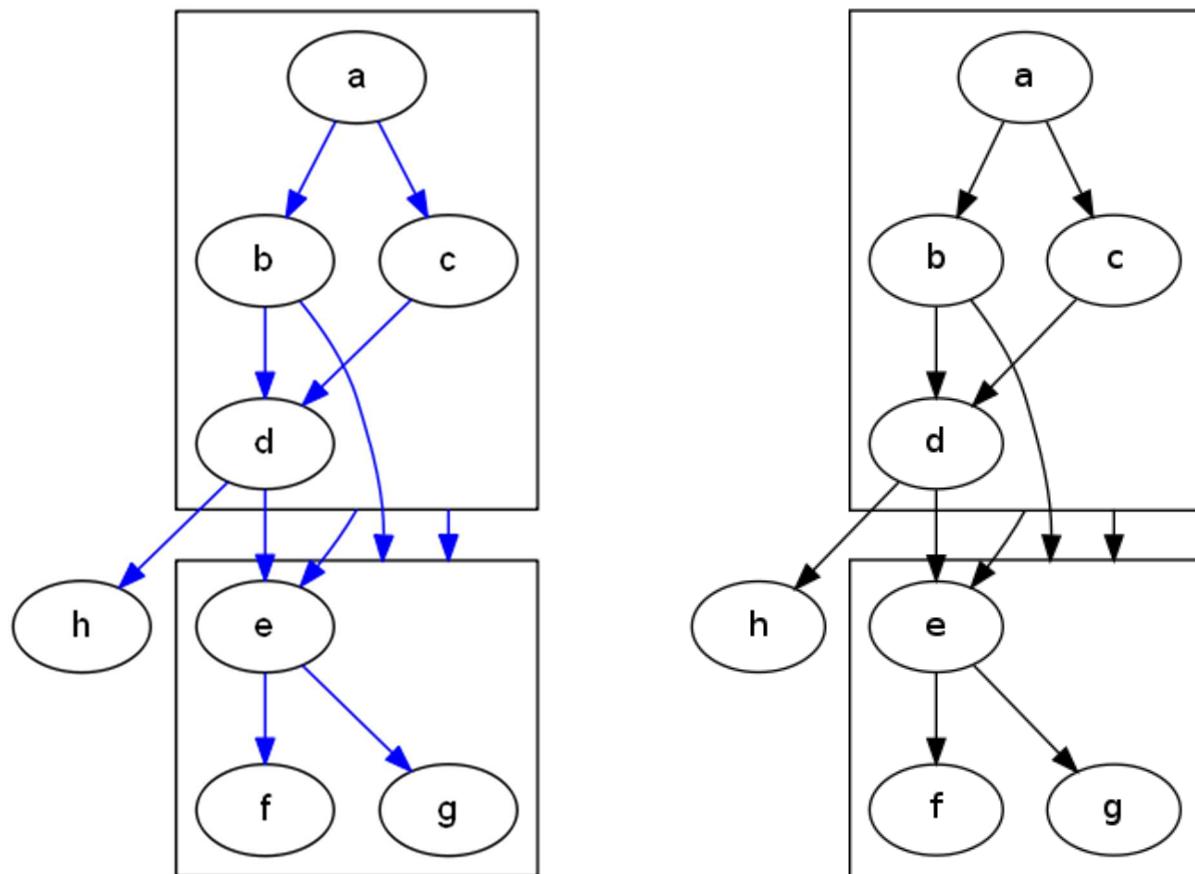
Press the "Refresh Graph" button and the graph now appears as:



This graph is almost the same as the graph from the Internet we are trying to duplicate. The only thing left to do is to remove the cluster labels that were included to illustrate this example. There is not a settings switch to turn these on/off, so here we must go back into the 'data' worksheet and delete the labels so that the data looks as follows:

8	#	cluster0
9	cluster0 {	
10	a	b
11	a	c
12	b	d
13	c	d
14	}	
15	#	cluster1
16	cluster1 {	
17	e	g
18	e	f
19	}	

Press the "Refresh Graph" button and the graph now appears as the graph on the left; the graph we are duplicating is on the right:



Generated by Relationship Visualizer

<http://i.stack.imgur.com/Ka0t2.png>

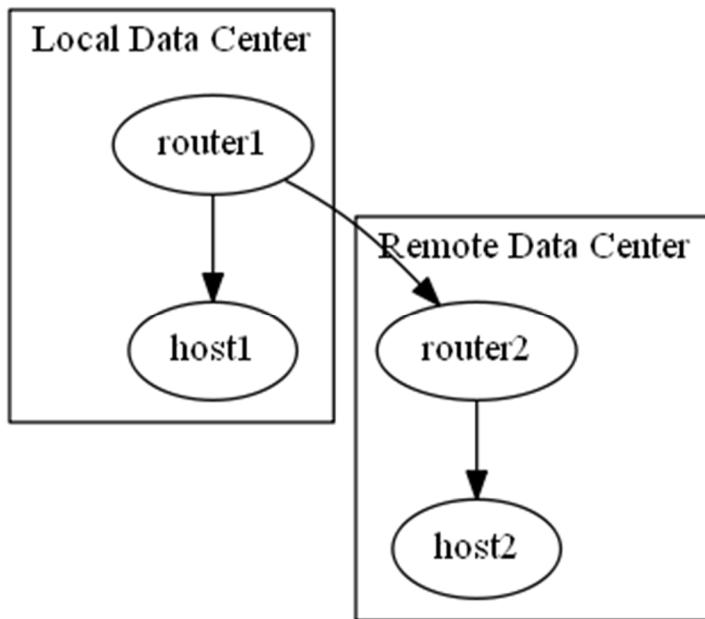
Excel to Graphviz for Win OS

The graph was purposely left with blue edges and a slightly smaller font size to differentiate it enough from the target graph to show it was not the original image but kept similar enough to show the goal was met.

If you are interested in making the final changes to make the graphs truly identical you can edit the style definition for the edge keyword on row 5 to remove the `color="blue"` attribute. Press the "Refresh Graph" button and the graph generated will be identical to the goal image.

Clusters - Aligning Nodes across Clusters

One of the ways Graphviz frees your time is by letting it choose the optimal way to lay out shapes and edges. Sometimes however you want to control the placement for esthetic reasons. Assume that you have the following graph:



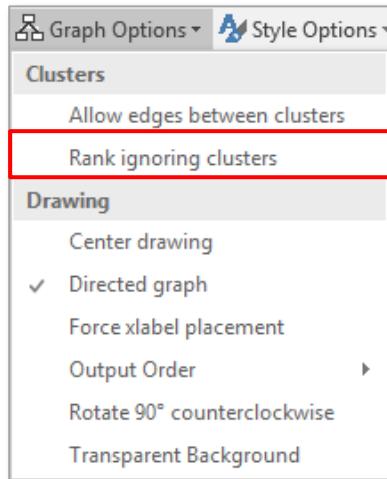
Created by this spreadsheet:

	A	B	C	G
1		Item	Label	Is Related To Item
2	#	Local Data Center		
3	{		Local Data Center	
4	router1			host1
5	}			
6	#	Remote Data Center		
7	{		Remote Data Center	
8	router2			host2
9	}			
10	#	Router to Router Link		
11	router1			router2

For esthetic reasons, we would like "router1" and "router2" to be aligned. Two native Graphviz DOT commands need to be added to the spreadsheet to make this occur. The first command is shown on line 4, which adds newrank="true" into the body of the main graph. (It appears that newrank is an undocumented attribute added in Graphviz 2.30 which activate a new ranking algorithm which allows defining rank="same" for nodes which belong to clusters).

A	B	C	G
1	Item	Label	Is Related To Item
2	# newrank allows defining rank=same for nodes which belong to clusters		
3	>	newrank="true"	

You can also achieve the same result by checking the graph option "Rank ignoring clusters"



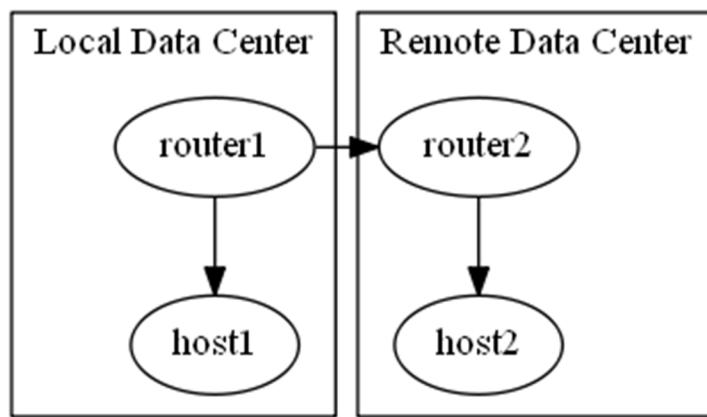
The next step is to add the following native Graphviz command after the cluster definitions:

```
{ rank="same"; "router1"; "router2"; }
```

as shown on row 13 below:

A	B	C	G
1	Item	Label	Is Related To Item
2	# newrank allows defining rank=same for nodes which belong to clusters		
3	>	newrank="true"	
4	# Local Data Center		
5	{	Local Data Center	
6	router1		host1
7	}		
8	# Remote Data Center		
9	{	Remote Data Center	
10	router2		host2
11	}		
12	# Align the router nodes		
13	>	{rank="same"; "router1"; "router2";;}	
14	# Router to Router Link		
15	router1		router2

Press the "Refresh Graph" button, and the graph contains router1 and router2 aligned as shown below:



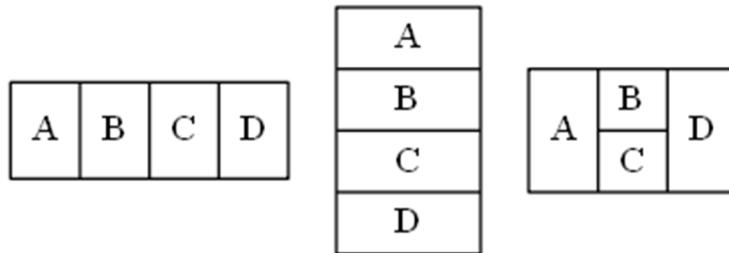
Nodes - Shape="Record"

Visually, a record is a box, with fields represented by alternating rows of horizontal or vertical sub-boxes. The Mrecord shape is identical to a record shape, except that the outermost box has rounded corners. Flipping between horizontal and vertical layouts is done by nesting fields in braces "...". The top-level orientation in a record is horizontal. Thus, a record with label "A | B | C | D" will have 4 fields oriented left to right, while "{A | B | C | D}" will have them from top to bottom and "A | {B | C} | D" will have "B" over "C", with "A" to the left and "D" to the right of "B" and "C".

The initial orientation of a record node depends on the rankdir attribute. If this attribute is "Top to Bottom" or "Bottom to Top", corresponding to vertical layouts, the top-level fields in a record are displayed horizontally. If, however, this attribute is "Left to Right" or "Right to Left", corresponding to horizontal layouts, the top-level fields are displayed vertically.

A	B	C	G	H	I
1	Item	Label	Is Related To Item	Style Name	Extra Style Attributes
2	node				shape="record"
3	# Left to right				
4	a	A B C D			
5	# Top to bottom				
6	b	{A B C D}			
7	# Mixed				
8	c	A {B C} D			

Results in:

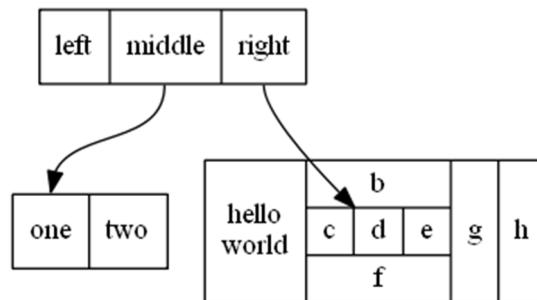


Nodes - Shape="Record" With Ports Specified

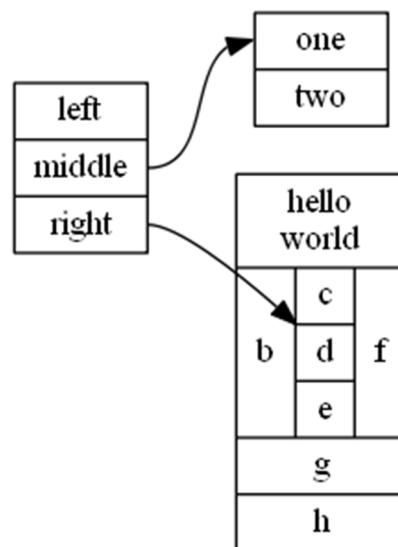
You can specify port identifiers as part of the field values in a record shape. The first string in the field Id assigns a port name to the field and can be combined with the node name to indicate where to attach an edge to the node.) The second string is used as the text for the field; it supports the usual escape sequences \n, \l and \r. Therefore, if we specify the following (with color-coding added to highlight the ports):

A	B	C	G	H	I
1	Item	Label	Is Related To Item	Style Name	Extra Style Attributes
2	node				shape="record"
3					
4	struct1	<f0> left <f1> middle <f2> right			
5	struct2	<f0>one <f1>two			
6	struct3	hello world { b { c <here> d e } f } g h			
7	struct1:f1		struct2:f0		
8	struct1:f2		struct3:here		

We will generate the following graph:



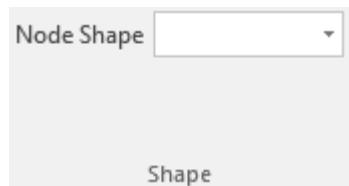
If we change the graphing direction from "Top to Bottom" to "Left to Right" on the 'Settings' worksheet and regenerate the graph it will appear as follows:



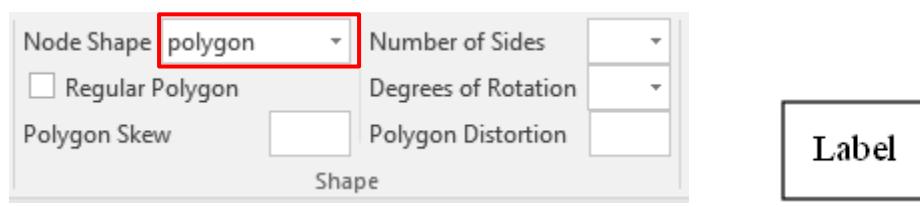
Nodes - Polygon Nodes

Polygon shapes are unique from other shapes in Graphviz and have extra attributes which control how the polygon is created.

The easiest way to learn about polygon shapes is to use the 'style designer'. The default 'Shape' section of the style designer ribbon (Element Type = Node) consists of a single dropdown list for "Node Shape". If you select 'polygon' as the shape the ribbon will change dynamically to present additional choices as shown below:

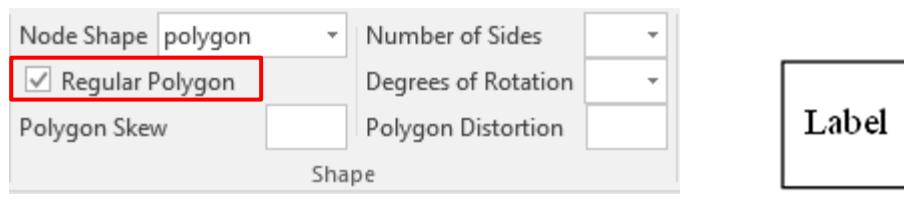


Selecting 'polygon' changes the ribbon to appear as:



shape="polygon"

Regular Polygon - If true, forces the polygon to be regular, i.e., the vertices of the polygon will lie on a circle whose center is the center of the node.

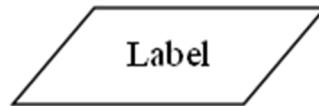


shape="polygon" regular="Yes"

Polygon Skew - Positive values skew top of polygon to right; negative values skew the top of the polygon to the left.

Positive Skew

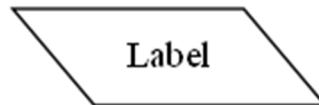
Node Shape	polygon	Number of Sides	
<input type="checkbox"/> Regular Polygon		Degrees of Rotation	
Polygon Skew	<input type="text" value="1"/>	Polygon Distortion	
Shape			



```
shape="polygon" skew="1" regular="No"
```

Negative Skew

Node Shape	polygon	Number of Sides	
<input type="checkbox"/> Regular Polygon		Degrees of Rotation	
Polygon Skew	<input type="text" value="-1"/>	Polygon Distortion	
Shape			



```
shape="polygon" skew="-1" regular="No"
```

Polygon Distortion - Positive values cause top part of the polygon to be larger than bottom; negative values do the opposite.

Positive Distortion

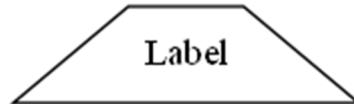
Node Shape	polygon	Number of Sides	
<input type="checkbox"/> Regular Polygon		Degrees of Rotation	
Polygon Skew		Polygon Distortion	<input type="text" value="1"/>
Shape			



```
shape="polygon" distortion="1" regular="No"
```

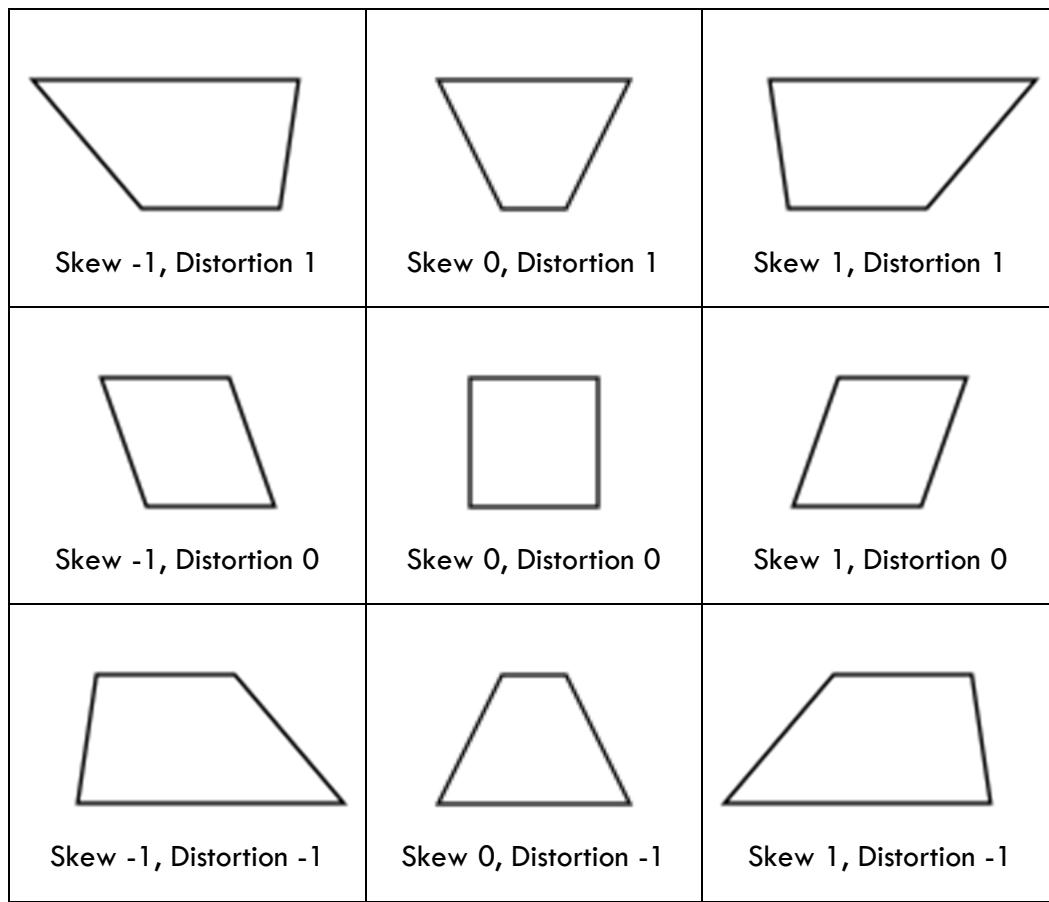
Negative Distortion

Node Shape	polygon	Number of Sides	
<input type="checkbox"/> Regular Polygon		Degrees of Rotation	
Polygon Skew		Polygon Distortion	<input type="text" value="-1"/>
Shape			



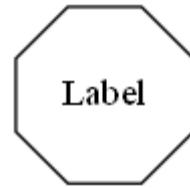
```
shape="polygon" distortion="-1" regular="No"
```

Combining Skew with Distortion



Number of Sides - Number of polygon sides.

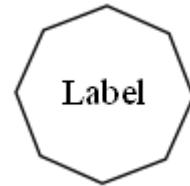
Node Shape <input type="button" value="polygon"/>	Number of Sides <input type="button" value="8"/>
<input checked="" type="checkbox"/> Regular Polygon	Degrees of Rotation <input type="button"/>
Polygon Skew <input type="button"/>	Polygon Distortion <input type="button"/>
Shape	



`shape="polygon" sides="8" regular="Yes"`

Polygon Rotation - Angle, in degrees, to rotate polygon node shapes. For any number of polygon sides, 0 degrees rotation results in a flat base.

Node Shape <input type="button" value="polygon"/>	Number of Sides <input type="button" value="8"/>
<input checked="" type="checkbox"/> Regular Polygon	Degrees of Rotation <input type="button" value="21°"/>
Polygon Skew <input type="button"/>	Polygon Distortion <input type="button"/>
Shape	



`shape="polygon" sides="8" regular="Yes" orientation="21"`

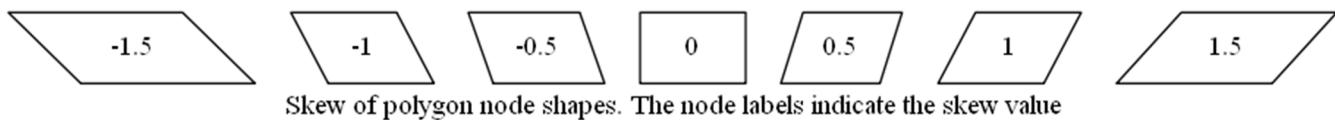
Nodes - Skewing the Angle of Polygon Nodes

Polygon shapes can be angled by changing the `skew` attribute. Positive values skew top of polygon to right; negative to left. This feature can be illustrated simply by defining seven nodes and varying the `skew` attribute by 0.5 for a range of values from -1.5 to 1.5 in the 'Attributes' column.

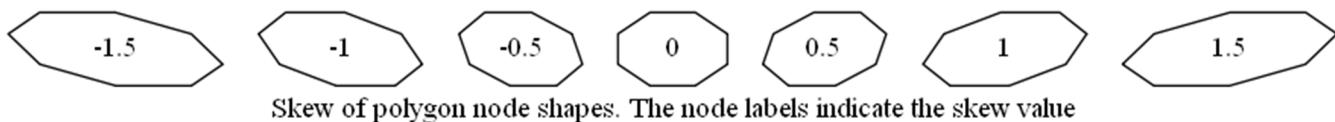
The Excel data is defined as follows:

A	B	C	G	H	I
1	Item	Label	Is Related To Item	Style Name	Extra Style Attributes
	graph	Skew of polygon node shapes. The node labels indicate the skew value			
2					
3	node				<code>shape="polygon" sides="4"</code>
4	1	-1.5			<code>skew="-1.5"</code>
5	2	-1.0			<code>skew="-1.0"</code>
6	3	-0.5			<code>skew="-0.5"</code>
7	4	0.0			<code>skew="0.0"</code>
8	5	0.5			<code>skew="0.5"</code>
9	6	1.0			<code>skew="1.0"</code>
10	7	1.5			<code>skew="1.5"</code>

Pressing the "Refresh Graph" button, the graph appears as:



This feature works for any number of polygon sides. If the `sides` attribute on row 3 changes from `sides="4"` to `sides="8"`, the resulting graph appears as:



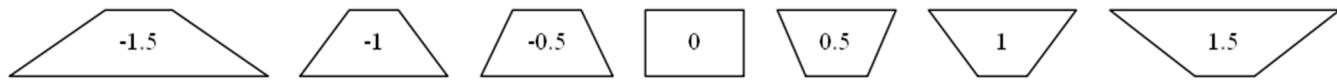
Nodes - Distorting the Length of Polygon Nodes

Polygon shapes can be distorted by changing the distortion attribute. Positive values cause the top part of the polygon to be larger than the bottom; negative values do the opposite. This feature can be illustrated simply by defining seven nodes and varying the distortion attribute by 0.5 for a range of values from -1.5 to 1.5 in the 'Attributes' column.

Define the Excel data as follows:

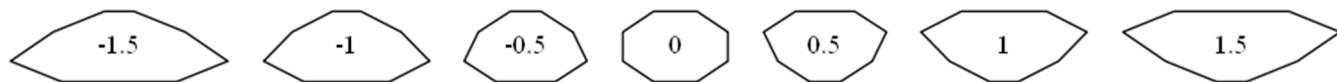
A	B	C	G	H	I
1	Item	Label	Is Related To Item	Style Name	Extra Style Attributes
2	graph	Distortion of polygon node shapes. The node labels indicate the distortion value			
3	node				
4	1	-1.5			shape="polygon" sides="4" distortion="-1.5"
5	2	-1.0			distortion="-1.0"
6	3	-0.5			distortion="-0.5"
7	4	0.0			distortion="0.0"
8	5	0.5			distortion="0.5"
9	6	1.0			distortion="1.0"
10	7	1.5			distortion="1.5"

Pressing the "Refresh Graph" button, the graph appears as:



Distortion of polygon node shapes. The node labels indicate the distortion value

This feature works for any number of polygon sides. If the sides attribute on row 3 changes from sides="4" to sides="8", the resulting graph appears as:



Distortion of polygon node shapes. The node labels indicate the distortion value

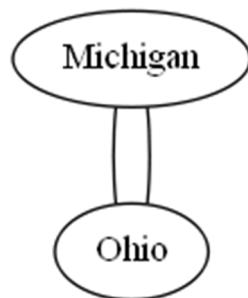
Edges - Consolidating Edges Using the 'strict' Option

Some sets of data will cause multiple edges to be drawn between the same nodes. A good example of this is the US state border example from the introduction of this document. Every state that shares a border with another state has two relationships. For example, Michigan shares a border with Ohio, and Ohio shares a border with Michigan.

Plotting the state border relationships as an undirected graph, the following data:

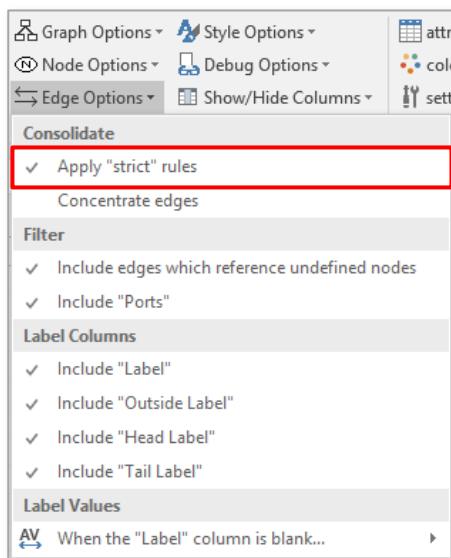
A	B	D	G
1	Item	Label	Related Item
2			
3	Michigan		Ohio
4	Ohio		Michigan

Generates the following graph:



Graphviz can consolidate these duplicate relationships into one relationship. If Graphviz is told that the graph is strict, then multiple edges are not allowed between the same pairs of nodes.

On the 'Graphviz' ribbon tab in the "Edge Options" section check the 'Apply "strict" rules' option to "yes" as shown below:



Press the "Refresh Graph" button and the graph appears as:



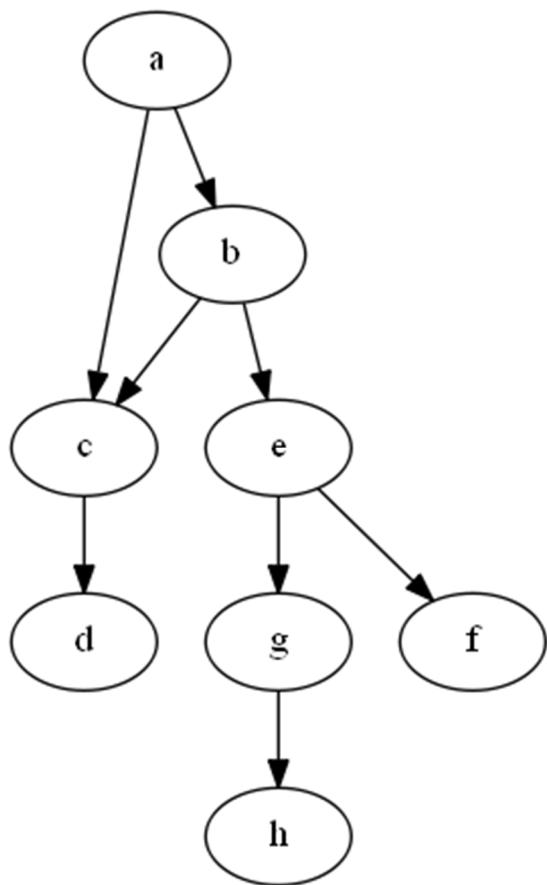
Edges - Changing the Order of Edges

Differences in style may be achieved by altering the edge ordering. If the value of the ordering attribute is "out", then the out edges of a node, that is, edges with the node as its tail node, must appear left-to-right in the same order in which they are defined in the input. If the value of the attribute is "in", then the in edges of a node must appear left-to-right in the same order in which they are defined in the input.

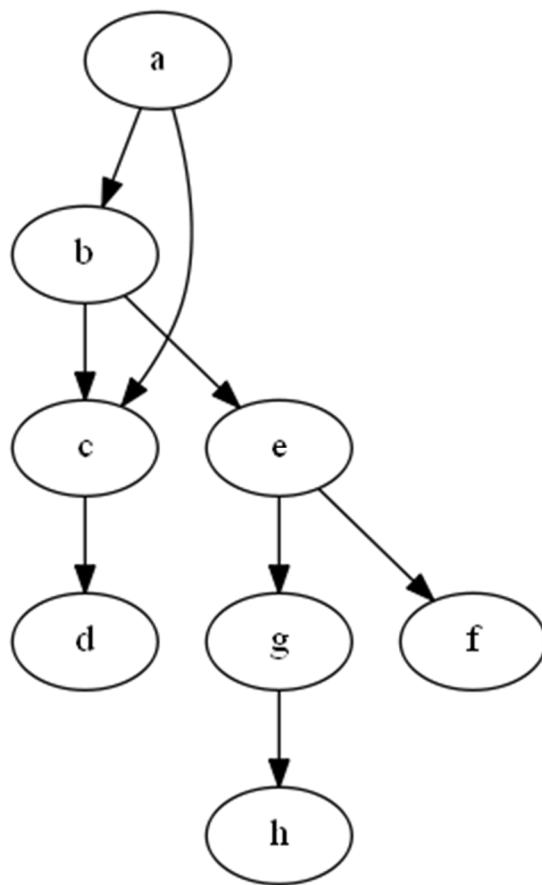
Assume you have several edge relationships defined as follows:

A	B	C	G	H	I
1	Item	Label	Is Related To Item	Style Name	Extra Style Attributes
2	graph				
3	a		b		
4	a		c		
5	b		c		
6	c		d		
7	b		e		
8	e		g		
9	e		f		
10	g		h		ordering="in"

Pressing the "Refresh Graph" button, the graph appears as:



ordering="in"



ordering="out"

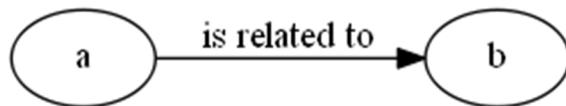
Edges - Placing a Label at the Head or Tail of an Edge

The default view of the Relationship Visualizer provides a Label column for labeling an edge. Graphviz also supports placing a label at the tail and/or head of an edge via the `taillabel` and `headlabel` attributes respectively. There are label columns on the 'data' worksheet which correspond to these attributes, however you must unhide the columns to use them.

For example, if we have a simple relationship such as:

	A	B	D	G
1	Item	Label	Related Item	
2				
3	a	is related to	b	

Producing the graph:



We can click the "Show/Hide Columns" dropdown list on the Graphviz tab to expose the additional label columns

Graph Options ▾ Style Options ▾
 Node Options ▾ Debug Options ▾
 Edge Options ▾ Show/Hide Columns ▾
 ✓ Comment Indicator
 ✓ Item
 Tail Label
 ✓ Label
 Outside Label
 Head Label
 ✓ Related Item
 Style Name
 Attributes
 Messages

Graph Options ▾ Style Options ▾
 Node Options ▾ Debug Options ▾
 Edge Options ▾ Show/Hide Columns ▾
 ✓ Comment Indicator
 ✓ Item
 ✓ Tail Label
 ✓ Label
 ✓ Outside Label
 ✓ Head Label
 ✓ Related Item
 Style Name
 Attributes
 Messages

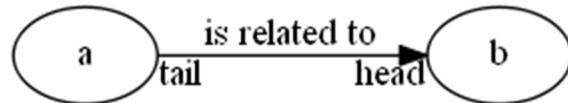
The data worksheet now appears as:

	A	B	C	D	E	F	G
1	Item	Tail Label	Label	Outside Label	Head Label	Related Item	
2							
3	a		is related to			b	

If we place the value "tail" at the tail of an edge, and "head" at the head of an edge, the data in the spreadsheet would look as follows:

	A	B	C	D	E	F	G
1	Item		Tail Label	Label	Outside Label	Head Label	Related Item
2							
3	a		tail	is related to		head	b

Pressing the "Refresh Graph" button creates the following graph:



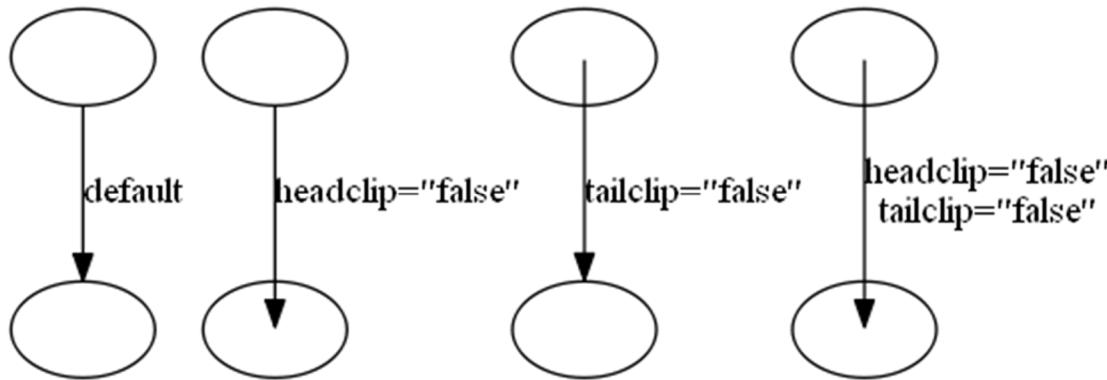
Edges - Drawing an Edge from or to the Center of a Node

An edge is clipped to the boundary of a node shape by default. It is possible to override this behavior such that the edge will begin and/or end in the center of the node instead of the node boundary. The attributes `headclip` and `tailclip` control this behavior. When set to true (the default) an edge is clipped to the boundary of a node. When set to false the edge goes to the center of the node, or the center of a port if applicable.

The data below:

A	B	C	G	H	I
1	Item	Label	Is Related To Item	Style Name	Extra Style Attributes
2	a,b,c,d,e,f,g,h				
3	a	default	b		
4	c	headclip="false"	d		headclip="false"
5	e	tailclip="false"	f		tailclip="false"
6	g	headclip="false"	h		headclip="false"
		tailclip="false"			tailclip="false"

Creates the following graph:

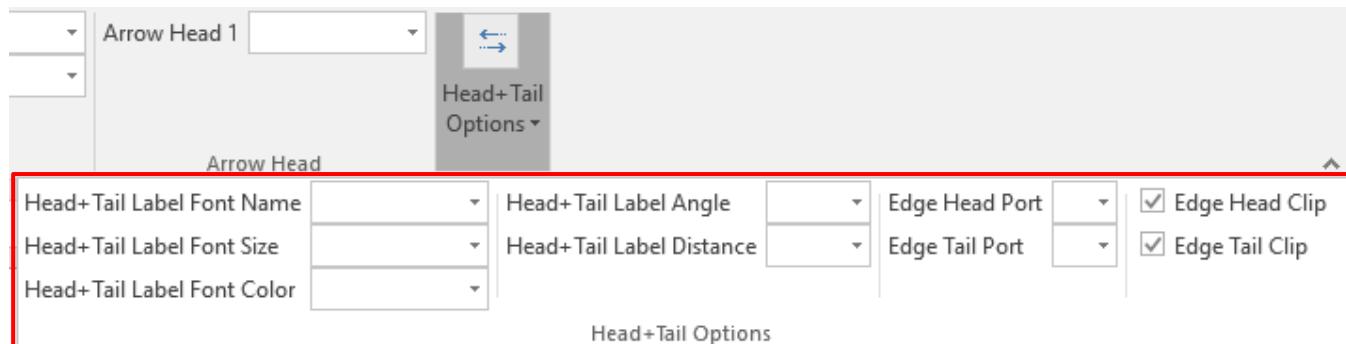


Note: The nodes have blank labels to make the illustration of edges coming from or going to the center of the node easier to see. Enabling the 'Nodes' graph option 'When the "Label" column is blank...''...use blank for the node label" on the 'Graphviz' ribbon tab is required to achieve this effect.

abl	When the "Label" column is blank...	...use blank for the node label
		✓ ...use the node identifier as the label

Edges - Head and Tail Options

The 'style designer' was updated in Version 5.0 to help provide more assistance in defining the head and tail attributes for an edge.



Head+Tail Label Font Name, Font Size, and Font Color - These attributes provide a way to differentiate the text at the end of the edges where they meet the node.



```
labelfontname="Arial" labelfontsize="8" labelfontcolor="Blue"
```

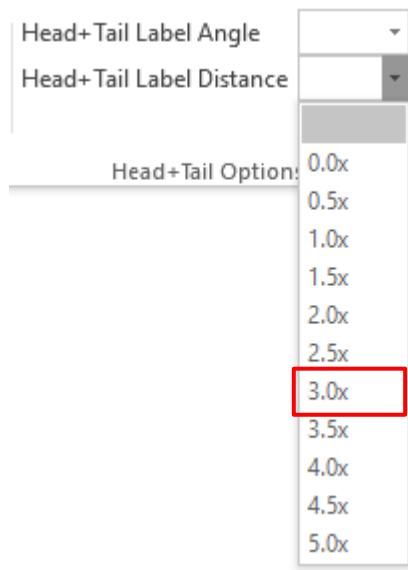
Label Angle - `labelangle=` (along with `labeldistance=`) , determines where the head label and tail label are placed with respect to the head (tail) in polar coordinates. The origin in the coordinate system is the point where the edge touches the node. The ray of 0 degrees goes from the origin back along the edge, parallel to the edge at the origin.

The angle, in degrees, specifies the rotation from the 0 degree ray, with positive angles moving counterclockwise and negative angles moving clockwise.



```
labelangle="90"
labelfontname="Arial" labelfontsize="8" labelfontcolor="Blue"
```

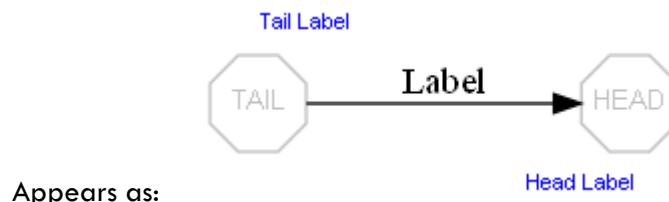
Label Distance - Multiplicative scaling factor adjusting the distance that the headlabel (taillabel) is from the head (tail) node. The default distance is 10 points. See labelangle for more details.



```

    labeldistance="3"
    labelfontname="Arial" labelfontsize="8" labelfontcolor="Blue"
  
```

Label Angle & Label Distance - This example depicts when labelangle= and labeldistance= attributes are used together.

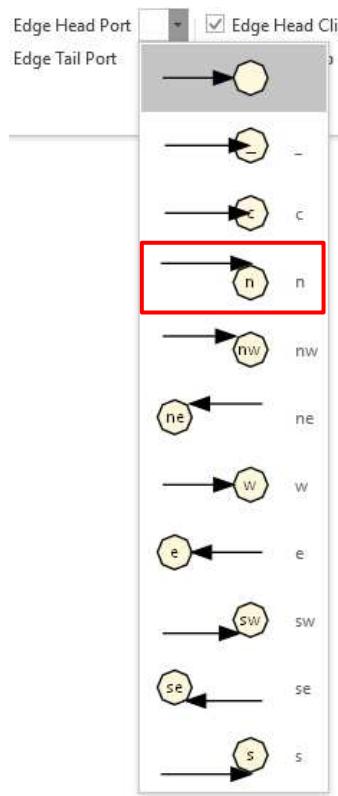


```

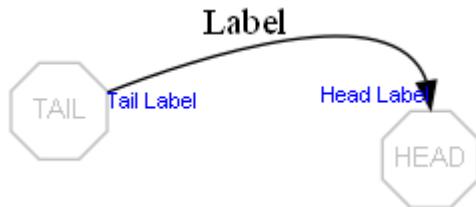
    labelangle="90" labeldistance="3"
    labelfontname="Arial" labelfontsize="8" labelfontcolor="Blue"
  
```

Edge Head Port - Indicates where on the head node to attach the head of the edge. In the default case, the edge is aimed towards the center of the node, and then clipped at the node boundary.

If a compass point is used, it must have the form "n", "ne", "e", "se", "s", "sw", "w", "nw", "c", "_". This specification modifies the edge placement to aim for the corresponding compass point on the port or, in the second form where no port name is supplied, on the node itself. The compass point "c" specifies the center of the node or port. The compass point "_" specifies that an appropriate side of the port adjacent to the exterior of the node should be used, if such exists. Otherwise, the center is used. If no compass point is used with a port name, the default value is "_".



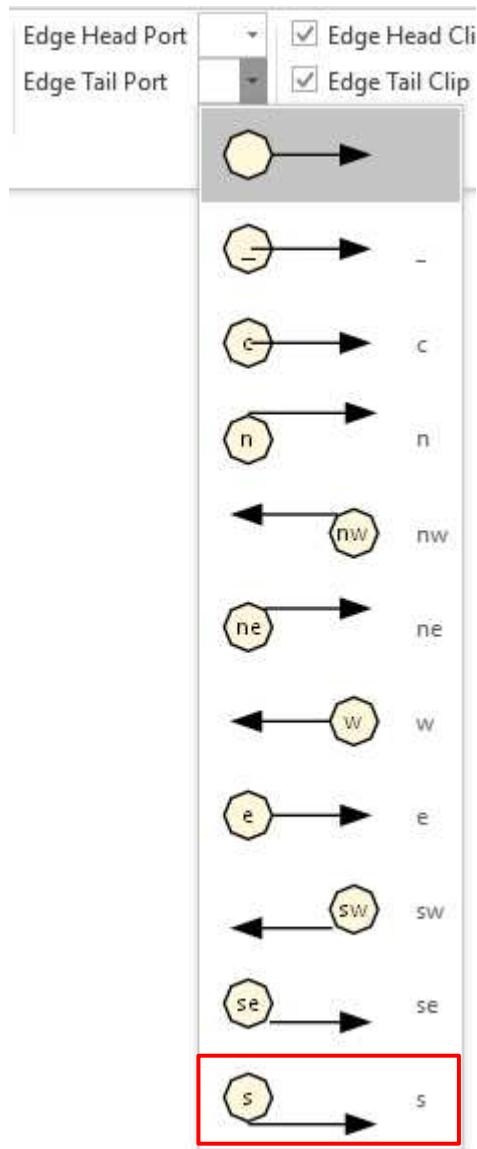
Appears As:



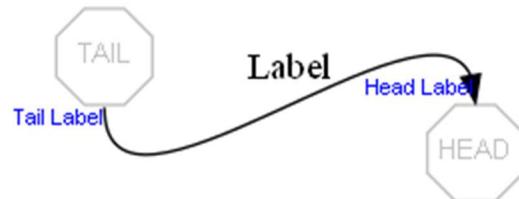
```
headport="n"
labelfontname="Arial" labelfontsize="8" labelfontcolor="Blue"
```

Edge Tail Port - Indicates where on the tail node to attach the tail of the edge.

If a compass point is used, it must have the form "n", "ne", "e", "se", "s", "sw", "w", "nw", "c", "_". This specification modifies the edge placement to aim for the corresponding compass point on the port or, in the second form where no port name is supplied, on the node itself. The compass point "c" specifies the center of the node or port. The compass point "_" specifies that an appropriate side of the port adjacent to the exterior of the node should be used, if such exists. Otherwise, the center is used. If no compass point is used with a port name, the default value is "_".



Appears as:



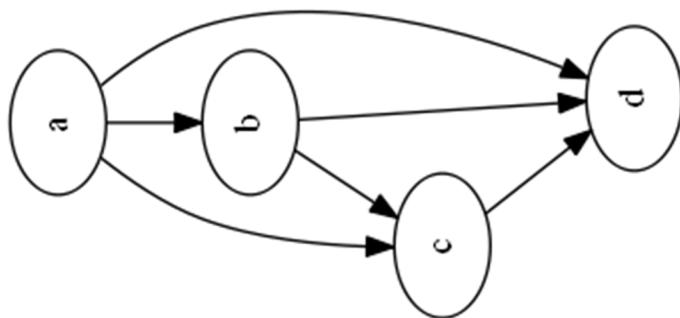
```
headport="n" tailport="s"
labelfontname="Arial" labelfontsize="8" labelfontcolor="Blue"
```

Graphs - Rotating Graphs 90 Degrees

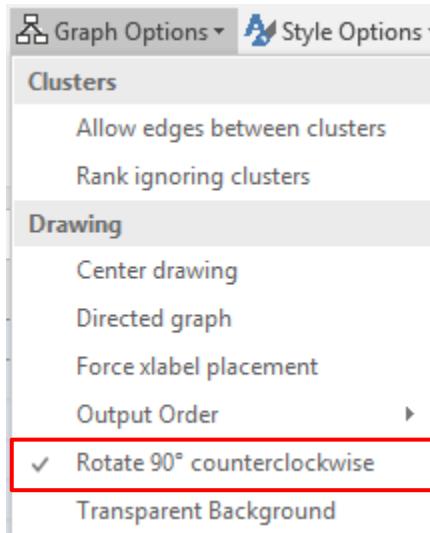
Graphs can have their drawing orientation set to landscape by setting the rotate attribute equal to 90. The final output is rotated in the counterclockwise direction. The data below:

A	B	C	G	H	I
1	Item	Label	Is Related To Item	Style Name	Extra Style Attributes
2	graph				
3	a		b		
4	a		c		
5	a		d		
6	b		c		
7	b		d		
8	c		d		

Creates the following graph:



An alternate method is to check the "Rotate 90 counter-clockwise" option from the Graph Options on the Graphviz tab.



USING SQL TO IMPORT DATA FROM OTHER EXCEL SPREADSHEETS

Have you ever wished that you could graph data that resides in some other spreadsheet? Do you have spreadsheets which change frequently, and thus need frequent re-graphing? The Relationship Visualizer provides a solution which lets you write Structured Query Language (SQL) statements which can pull in your data from other worksheets in an easily repeatable manner. Using this capability is optional. You do not have to use this feature to create graphs. This is a feature for the Relationship Visualizer power user and requires some working knowledge of how to write SQL statements.

SQL Explained

Structured Query Language (SQL) is a domain-specific language designed for stream processing in a relational data stream management system (RDSMS). It is particularly useful in handling structured data where there are relations between different entities/variables of the data.

SQL SELECT statements are designed for a specific purpose: to query data contained in a relational database. The SQL SELECT statement selects data from a database, specifying column(s) headings, table(s) from which columns are to be selected, and the criteria for selection. SELECT * is used to select all columns from a table. The FROM clause in a SELECT statement specifies the Table from which columns are to be selected. The WHERE clause in a SELECT statement specifies the criteria for selection.

Syntax: `SELECT column_name FROM table_name [WHERE clause]`

ActiveX Data Objects (ADO) SQL Queries

Excel can function as a flat-file RDBMS database that can be accessed using ActiveX Data Objects (ADO) and SQL queries. Microsoft Excel handles SQL via its own SQL dialect. See the references at the end of this chapter for detailed information on the Excel SQL dialect.

The Relationship Visualizer contains a worksheet named 'sql' which hides most of the complexities of writing ADO SQL. It allows you to write SQL SELECT queries conforming to the ADO syntax conventions and it takes care of the rest. The SQL SELECT statements can reference either the Relationship Visualizer spreadsheet or another Excel spreadsheet as the source of the data. When the query is executed, the query results are written to the 'data' worksheet ready for graphing.

ADO has a lot of powerful features, but the only features which the Relationship Visualizer uses are SQL SELECT queries of the form:

```
SELECT [Country Code] as [Item],  
      [Country Name] as [Label]  
FROM   [Countries$]
```

This is a standard SQL query, one that selects two columns in the database (which in our case is a worksheet) and names the results to correspond to the column names of the 'data' worksheet. Notice that we specify the name of an individual worksheet in the query, in the same way that we would specify an individual table name if we were connecting to a database, with a couple of minor exceptions. The worksheet name is enclosed in square brackets and that the actual worksheet name - **Countries** - has a \$ appended to it.

Square brackets are also used to wrap the column names since the column names may have spaces present. The square braces inform ADO to treat the value inside as the column name. Square brackets are used instead of quotes because quotes denote a string value to place in the results.

The results of the SELECT statements are processed by the Relationship Visualizer to place the selected values into the appropriate columns of the 'data' worksheet. The processing logic only recognizes the column names, and will discard all other data:

- [Comment]
- [Item]
- [Label]
- [Outside Label]
- [Tail Label]
- [Head Label]
- [Related Item]
- [Style Name]
- [Attributes]

Excel SQL also allows you to specify strings values as selection columns. The string is repeated for each row of the query results when the SQL is executed. This is a useful feature for specifying things such as a style name. Assume that there is a style defined on the 'styles' worksheet named "Country". The query is modified as follows to add a style named "Country" to each row of the results:

```
SELECT [Country Code] as [Item],
       [Country Name] as [Label],
       'Country'      as [Style Name]
  FROM   [Countries$]
```

SQL allows you to specify WHERE conditions which control what information is returned. The WHERE clause, states the qualifying conditions for a query. Multiple conditions can be joined by the AND and OR clauses, optionally surrounded by (parentheses) to group them. Only the records that satisfy the specified criteria are returned by the query.

Assume that we wanted a list of countries which use US Dollars as their national currency. The international code for US Dollars is "USD" so the SQL would be modified to appear as follows:

```
SELECT [Country Code] as [Item],
       [Country Name] as [Label],
       'Country'      as [Style Name]
  FROM   [Countries$]
 WHERE  [Currency Code] = 'USD'
```

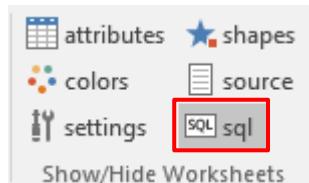
A column can contain duplicate values, and to list the distinct values, use the SELECT DISTINCT clause. The DISTINCT clause can be used to return only distinct values from a set of records. For example, if our worksheet contains the continent names where countries reside, we can obtain a list of the continents without duplicate rows using the following SQL statement:

Assume that we wanted a list of countries which use US Dollars as their national currency. The international code for US Dollars is "USD" so the SQL would be modified to appear as follows:

```
SELECT DISTINCT [Continent] as [Item],
               [Continent] as [Label],
               'Continent' as [Style Name]
FROM          [Countries$]
```

Now that you understand the basics of writing SQL SELECT statements, let us look at the features contained in the 'sql' worksheet and ribbon tab.

The 'sql' worksheet is hidden by default. To expose the 'sql' worksheet select the 'sql' button on the 'Graphviz' tab.



SQL Ribbon Controls

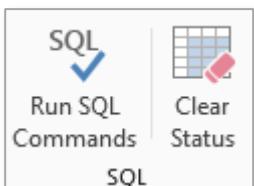
The 'SQL ribbon tab is activated whenever the 'sql' worksheet is activated. It appears as follows:



It contains the following major groups:

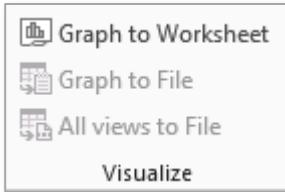
- SQL
- Visualize

SQL



Label	Control Type	Description
Run SQL Commands	Button	Sequentially runs the SQL statements in this worksheet, writing the results to the 'data' worksheet.
Clear Status	Button	Clear the values from the 'Status' column.

Visualize



Label	Control Type	Description
Refresh Graph	Button	Graphs the data in the 'data' worksheet and displays the results in Excel. This button does the same work as the like-named button on the 'Graphviz' ribbon tab and is included here to speed the creation of graphs after running SQL statements.
Graph to File	Button	Graphs the data in the 'data' worksheet and writes the graph to a file using the values specified on the Graphviz ribbon tab. This button does the same work as the like-named button on the 'Graphviz' ribbon tab and is included here to speed the creation of graphs after running SQL statements.
All views to File	Button	Graphs the data in the 'data' worksheet against all the views specified and writes the graphs to files using the values specified on the Graphviz ribbon tab. This button does the same work as the like-named button on the 'Graphviz' ribbon tab and is included here to speed the creation of graphs after running SQL statements.

Writing SQL Queries to create graphs

In the Relationship Visualizer download zip file, there is a directory named "SQL - World Map" within the "samples" directory. We will use this example to show how to use the 'sql' worksheet. The "SQL - World Map" directory has in it a copy of the Relationship Visualizer spreadsheet containing SQL statements, and predefined styles on the 'styles' worksheet, as well as 'data' directory which has 5 representations of a "countries" worksheet corresponding to the supported file extensions of csv, xls, xlsx, xlsm, and xslx. All of the "countries" spreadsheets have the same data. The query logic has to treat each file extension a little differently (I won't bore you with the details) and is included in the different formats simply for testing purposes. An example of a "countries" workbook appears as follows:

Excel to Graphviz for Win OS

The screenshot shows a Microsoft Excel spreadsheet titled "countries.xlsx - Excel". The table contains data for 246 countries, including ISO codes, names, populations, areas, and continent codes. The columns are labeled A through O. The "Continent" column uses abbreviations like "Europe", "Asia", "Africa", etc. The "Currency Code" column includes codes like "EUR", "USD", "AOA", etc. The "Phone" column has various formats. The "Postal Code Format" column contains formulas such as `^(\d{2})AD` and `^(\d{6})`. The "Postal" column also contains formulas like `^(\d{2})AD` and `^(\d{6})`.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
1	ISO	ISO3	ISO-Numeric	tips	Country	Capital	AreaInSquareKm	Population	ContinentCode	Continent	tld	Currency Code	Currency Name	Phone	Postal Code Format	Postal
2	AD	AND	20	AN	Andorra	Andorra la Vella	468	84000	EU	Europe	.ad	EUR	Euro	376 AD##	^(\d{2})AD	
3	AE	ARE	784	AE	United Arab Emirates	Abu Dhabi	82880	4975953	AS	Asia	.ae	AED	Dirham	971		
4	AF	AFG	4	AF	Afghanistan	Kabul	647500	29121286	AS	Asia	.af	AFN	Afghani	93		
5	AG	ATG	28	AC	Antigua and Barbuda	St. John's	443	86754	NA	North America	.ag	XCD	Dollar	-267		
6	AI	AIA	660	AV	Anguilla	The Valley	102	13254	NA	North America	.ai	XCD	Dollar	-263		
7	AL	ALB	8	AL	Albania	Tirana	28748	2986952	EU	Europe	.al	ALL	Lek	355		
8	AM	ARM	51	AM	Armenia	Yerevan	29800	2986000	AS	Asia	.am	AMD	Dram	374 #####	^(\d{6})	
9	AO	AGO	24	AO	Angola	Luanda	1246700	13068161	AF	Africa	.ao	AOA	Kwanza	244		
10	AQ	ATA	10	AY	Antarctica		14000000	0	AN	Antarctica	.aq					
11	AR	ARG	32	AR	Argentina	Buenos Aires	2768890	41343201	SA	South America	.ar	ARS	Peso	54 @####@@@	^([A-Z])	
12	AS	ASM	16	AQ	American Samoa	Pago Pago	199	57881	OC	Oceania	.as	USD	Dollar	-683		
13	AT	AUT	40	AU	Austria	Vienna	83858	8205000	EU	Europe	.at	EUR	Euro	45 ####	^(\d{4})	
14	AU	AUS	36	AS	Australia	Canberra	7688850	21515754	OC	Oceania	.au	AUD	Dollar	61 ####	^(\d{4})	
15	AW	ABW	533	AA	Aruba	Oranjestad	193	71566	NA	North America	.aw	AWG	Guilder	297		
16	AX	ALA	248	AL	Aland Islands	Mariehamn	26711	111	EU	Europe	.ax	EUR	Euro	340		
17	AZ	AZE	31	AJ	Azerbaijan	Baku	86600	830512	AS	Asia	.az	AZN	Manat	994 AZ ####	^([A-Z])	
18	BA	BH	70	BK	Bosnia and Herzegovina	Sarajevo	51129	4590000	EU	Europe	.ba	BAM	Marka	387 #####	^(\d{5})	
19	BB	BRB	52	BB	Barbados	Bridgetown	431	285653	NA	North America	.bb	BBD	Dollar	-245 BB####	^([A-Z])	
20	BD	BGD	50	BG	Bangladesh	Dhaka	144000	156118464	AS	Asia	.bd	BDT	Taka	880 ####	^(\d{4})	
21	BE	BEL	56	BE	Belgium	Brussels	30510	10403000	EU	Europe	.be	EUR	Euro	32 ####	^(\d{4})	
22	BF	BFA	854	UV	Burkina Faso	Ouagadougou	274200	16241811	AF	Africa	.bf	XOF	Franc	226		
23	BG	BGR	100	BU	Bulgaria	Sofia	110910	7147875	EU	Europe	.bg	BGN	Lev	359 ####	^(\d{4})	
24	BH	BHR	48	BA	Bahrain	Manama	665	738004	AS	Asia	.bh	BHD	Dinar	973 #####	^(\d{3})	
25	BI	BDI	108	BY	Burundi	Bujumbura	27830	9863117	AF	Africa	.bi	BIF	Franc	257		
26	BJ	BEN	208	BN	Benin	Porto-Novo	112620	9056010	AF	Africa	.bj	XOF	Franc	229		
27	BL	BLM	652	TB	Saint Barthelemy	Gustavia	21	8450	NA	North America	.gp	EUR	Euro	590 #### ##		
28	BM	BMU	60	BD	Bermuda	Hamilton	53	65365	NA	North America	.bm	BMD	Dollar	-440 @@ ##	^([A-Z])	
29	BN	BRN	96	BX	Brunei	Bandar Seri Begawan	5770	395027	AS	Asia	.bn	BND	Dollar	673 @@####	^([A-Z])	
30	BO	BOL	68	BL	Bolivia	Sucre	1098580	9947418	SA	South America	.bo	BOB	Boliviano	591		
31	BQ	BES	535	BS	Bonaire, Saint Eustatius and Saba		18012	NA	North America	.bq	USD	Dollar	599			
32	BR	BRA	76	BR	Brazil	Brasilia	8511965	20110330	SA	South America	.br	BRL	Real	55 ##### ##	^(\d{8})	
33	BS	BHS	44	BF	Bahamas	Nassau	13940	301790	NA	North America	.bs	BSD	Dollar	-241		
34	BT	BTN	64	BT	Bhutan	Thimphu	47000	699847	AS	Asia	.bt	BTN	Ngultrum	975		
35	BV	BVT	74	BV	Bouvet Island		0	AN	Antarctica	.bv	NOK	Krone				
36	BW	BWA	72	BC	Botswana	Gaborone	600370	2029307	AF	Africa	.bw	BWP	Pula	267		
37	BY	BLR	112	BO	Belarus	Minsk	207600	9685000	EU	Europe	.by	BYR	Ruble	375 #####	^(\d{6})	
38	CA	CAN	148	CA	Canada	Ottawa	20665	31456154	NA	North America	.ca	CAD	Dollar	601		

Now open the file "Relationship Visualizer.xslm", select the 'sql' worksheet. Note that there are 4 rows with SQL statements. We will use these statements to extract the values out the 'countries' workbooks and draw a graph view of the continents, countries, and connected borders. The 'sql' worksheet appears as follows:

The screenshot shows the "SQL" worksheet of the "Relationship Visualizer.xslm" Excel file. It contains four rows of SQL code intended for generating data from the "countries" workbook. The first three rows use the "countries.xlsx" file, while the fourth row uses "countries.xls". The statements involve selecting data from tables like "countries", "continents", and "neighbours" based on ISO codes and continent names.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	SQL SELECT Statement		Excel data file (.xlsx, .xslm, .xlsb, or .xls only)	Status											
2	SELECT [ISO] as [Item], [ISO] as [Label], [Country] as [Outside Label], [Continent] as [Style Name]		countries.xlsx												
3	FROM [countries\$]														
4	WHERE [ISO] IS NOT NULL														
5	ORDER BY [Country] ASC														
6															
7															
8															
9															
10															

The first SQL statement on row 2 selects the column "ISO" to represent the Item ID, as well as the node label. The country name is selected to be an outside label, and the continent name is selected to be the style name. Within the 'styles' worksheet there are 7 node style definitions which correspond to the continent names.

The SQL is written as follows:

```
SELECT      [ISO]           as [Item],
            [ISO]           as [Label],
            [Country]       as [Outside Label],
            [Continent]     as [Style Name]
FROM        [countries$]
WHERE       [ISO] IS NOT NULL
ORDER BY    [Country] ASC
```

The second SQL statement on row 3 will extract the 7 seven unique continent names from the list of countries by using the DISTINCT clause. The continent name will be used as the Item ID as well as the node label. A style name of "Continent" will be used for all the rows.

The SQL is written as follows:

```
SELECT DISTINCT [Continent]   as [Item],
               [Continent]   as [Label],
               'Continent'   as [Style Name]
FROM        [countries$]
ORDER BY    [Continent] ASC
```

The third SQL statement on row 4 creates edge relationship rows by selecting the ISO value as the Item ID, and the "Neighbors" column as the "Related Item" value. The "Neighbors" column in the source worksheet contains comma-delimited ISO values which are neighboring countries to that row. The Relationship Visualizer has built-in logic to expand the comma-delimited list into multiple relationships. No style name is provided, so the default edge style will be used. Note that on the WHERE clause the directive "IS NOT NULL" has been added. This clause causes the query to skip rows with empty cells, as there are no values with which to express relationships.

The SQL is written as follows:

```
SELECT      [ISO]           as [Item],
            [Neighbors]     as [Related Item]
FROM        [countries$]
WHERE       [Neighbors] IS NOT NULL
```

The fourth SQL statement on row 5 is used to group countries by continent. It creates edge relationships by placing the Continent name as the Item ID, and the ISO country code in the "Related Item" column. The style name is specified as "ContinentToCountry".

The SQL is written as follows:

Excel to Graphviz for Win OS

```
SELECT      [Continent]          as [Item],  
            [ISO]                as [Related Item],  
            'ContinentToCountry' as [Style Name]  
FROM        [countries$]  
WHERE       [ISO] IS NOT NULL
```

Each of these queries have to specify the name of the workbook against which the SQL is to run. This information is contained in column C. There are three ways the file name may be specified.

1. Specify the full path to the file as shown in the sample file.
2. Specify just the file name. The Relationship Visualizer will assume that the data workbook is contained in the same directory as the Relationship Visualizer workbook.
3. Leave the cell blank. If the cell is blank the Relationship Visualizer assumes that the data is contained within the current Relationship Visualizer workbook.

Once the file paths are specified, we can run the SQL. Press the "Run SQL Commands" button



The SQL commands are run in sequence from top to bottom. Results are written to the 'data' worksheet, and the query result status is displayed in column D such as:

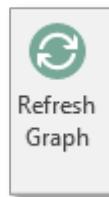
The screenshot shows an Excel spreadsheet titled "World Map.xlsx - Excel". The ribbon menu is visible at the top, and the "SQL" tab is selected. The main area contains five rows of data, each representing a SQL statement. Column A contains the SQL code, column B contains the target file name, and column C contains the status of the execution. All five statements show a "SUCCESS" status in column D.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	SQL SELECT Statement		Excel data file (.xlsx, .xism, .xlsb, or .xls only)	SUCCESS											
1	SELECT [ISO] as [Item], [ISO] as [Label], [Country] as [Outside Label], [Continent] as [Style Name] FROM [countries\$] WHERE [ISO] IS NOT NULL ORDER BY [Country] ASC		countries.xlsm	SUCCESS											
2	SELECT DISTINCT [Continent] as [Item], [Continent] as [Label], 'Continent' as [Style Name] FROM [countries\$] ORDER BY [Continent] ASC		countries.xlsx	SUCCESS											
3	SELECT [ISO] as [Item], [Neighbors] as [Related Item] FROM [countries\$] WHERE [Neighbors] IS NOT NULL		countries.xlsb	SUCCESS											
4	SELECT [Continent] as [Item], [ISO] as [Related Item], 'ContinentToCountry' as [Style Name] FROM [countries\$] WHERE [ISO] IS NOT NULL		countries.xls	SUCCESS											
5															
6															
7															
8															
9															
10															

If we switch worksheets to the 'data' worksheet, it appears as follows:

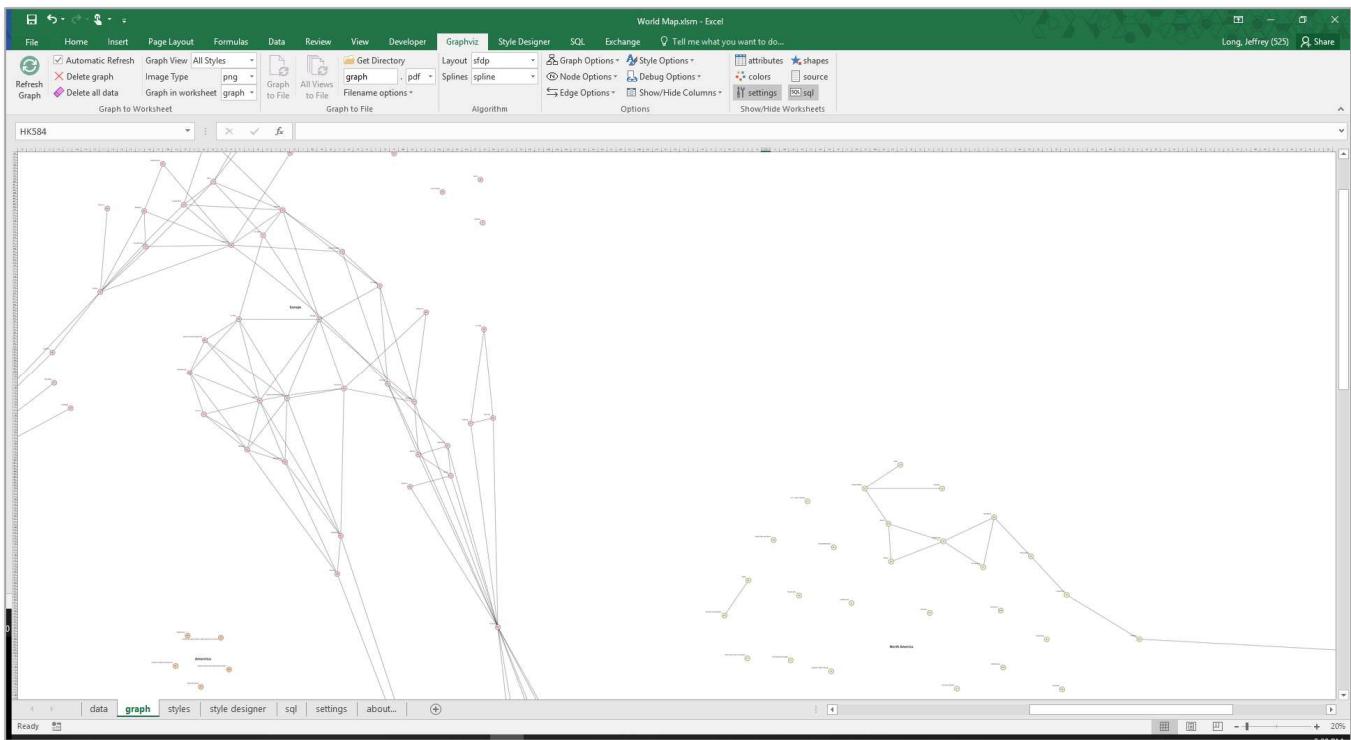
Item	Label	Related Item	Style Name	Attributes
AF	AF		Asia	
AX	AX		Europe	
AL	AL		Europe	
DZ	DZ		Africa	
AS	AS		Oceania	
AD	AD		Europe	
AO	AO		Africa	
AI	AI		North America	
AQ	AQ		Antarctica	
AG	AG		North America	
AR	AR		South America	
AM	AM		Asia	
AW	AW		North America	
AU	AU		Oceania	
AT	AT		Europe	
AZ	AZ		Asia	
BS	BS		North America	
BH	BH		Asia	
BD	BD		Asia	
BB	BB		North America	
BY	BY		Europe	
BE	BE		Europe	
BZ	BZ		North America	
BJ	BJ		Africa	
BM	BM		North America	
BT	BT		Asia	
BO	BO		South America	
BQ	BQ		North America	

The data is all present and in the appropriate columns for graphing. In this example 678 rows of data have been created using 4 SQL statements! Press the "Refresh Graph" button to graph the data. Since this is a large data set, be prepared to wait a little while for the results.



When Graphviz completes its work, you should see a logical world graph which appears as follows (sent to 'graph' worksheet with zoom level set to 30%):

Excel to Graphviz for Win OS



For more information on Excel SQL

- Writing SQL Queries against Excel files (Excel SQL) - <https://querysurge.zendesk.com/hc/en-us/articles/205766136-Writing-SQL-Queries-against-Excel-files>
- SQL Query Reference - <https://www.aquaclusters.com/app/home/project/public/aquadatastudio/wikibook/MS-Excel/page/SQL-Query-Reference/SQL-Query-Reference>
- Fundamental Microsoft Jet SQL for Access 2000 - <https://docs.microsoft.com/en-us/previous-versions/office/developer/office2000/aa140011%28v%3doffice.10%29>
- Intermediate Microsoft Jet SQL for Access 2000 - <https://docs.microsoft.com/en-us/previous-versions/office/developer/office2000/aa140015%28v%3doffice.10%29>

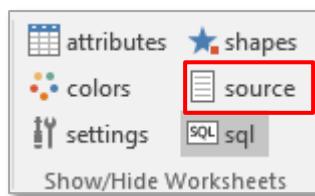
WORKING WITH THE DOT LANGUAGE

The Relationship Visualizer is intended to remove the need to learn the DOT language but still create Graphviz graphs. It can, however, also be a very useful tool for learning the DOT language. This chapter will explain how you can access the DOT source code which is generated from the Excel worksheets and passed to the Graphviz layout engine. These capabilities allow you to graph relationships, and then see the underlying code.

Viewing DOT Source Code

The Relationship Visualizer contains a worksheet named 'source' which shows the source code which was generated whenever a graphing button is pressed.

The 'source' worksheet is hidden by default. To expose the 'source' worksheet select the 'source' button on the 'Graphviz' tab.



Click on the 'source' worksheet to make it the active worksheet. The 'Source' ribbon tab is activated whenever the 'source' worksheet is activated. It appears as follows:

A screenshot of the Microsoft Excel application window titled 'Relationship Visualizer.xlsx - Excel'. The ribbon is set to the 'Source' tab. The worksheet 'source' is active, showing the following content:

```

Line 1: Line 1 Graphviz Source
Line 2:
Line 3:
Line 4:
Line 5:
Line 6:
Line 7:
Line 8:
Line 9:
Line 10:
Line 11:
Line 12:
Line 13:
Line 14:
Line 15:
Line 16:
Line 17:
Line 18:
Line 19:
Line 20:
Line 21:
Line 22:
Line 23:
Line 24:
Line 25:
Line 26:
Line 27:
Line 28:
Line 29:
Line 30:

```

The status bar at the bottom shows 'Ready'.

Excel to Graphviz for Win OS

You are probably thinking "but I don't see any DOT source code". That is correct. Whenever the "Refresh Graph" or "Graph to File" button is pressed the source code will be present.

Consider the following example:

Screenshot of Microsoft Excel showing a relationship visualization between items. The data is displayed in a table and a corresponding graph is generated on the right.

Data Table:

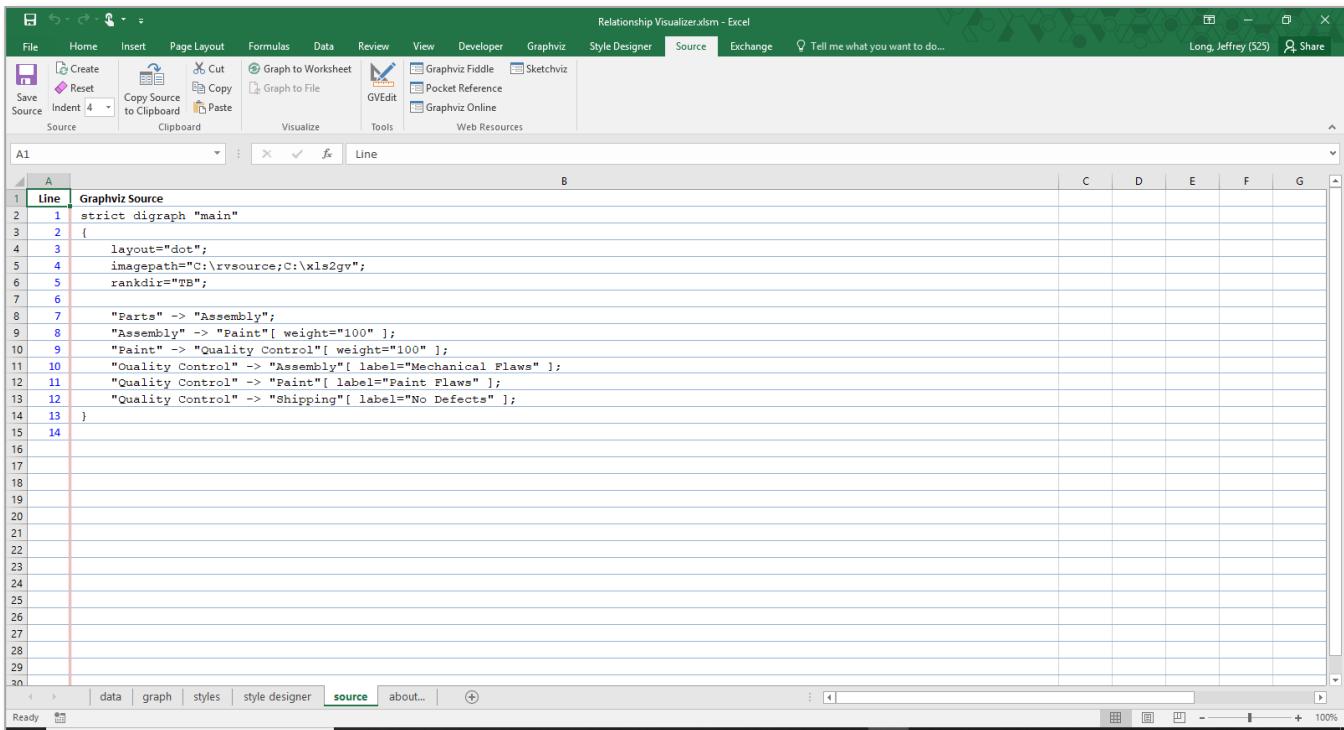
	A	B	D	G	I	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	Item	Label		Related Item	tributes															
2																				
3	Parts			Assembly																
4	Assembly			Paint																
5	Paint			Quality Control																
6	Quality Control			Mechanical Flaws																
7	Quality Control			Paint Flaws																
8	Quality Control			No Defects																
9				Shipping																
10																				
11																				
12																				
13																				
14																				
15																				
16																				
17																				
18																				
19																				
20																				
21																				
22																				
23																				
24																				
25																				
26																				
27																				
28																				
29																				
30																				

Graph:

```
graph TD; Parts((Parts)) --> Assembly((Assembly)); Assembly --> Paint((Paint)); Paint --> QualityControl((Quality Control)); QualityControl --> Shipping((Shipping)); QualityControl -- "Mechanical Flaws" --> Paint;
```

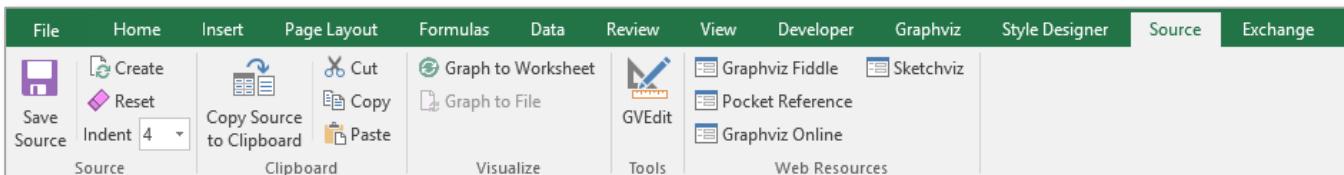
The graph shows a flow from Parts to Assembly, Assembly to Paint, Paint to Quality Control, and Quality Control to Shipping. There is also a feedback loop from Quality Control back to Paint labeled "Mechanical Flaws".

To see the Graphviz source code switch to the 'source' worksheet. It appears as:



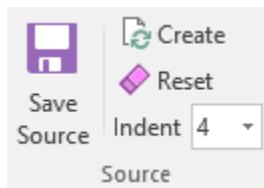
The "Source" Ribbon Tab

Now that you understand the basics of viewing Graphviz source code, let us look at the features contained in the 'Source' ribbon tab. The 'Source' ribbon tab is activated whenever the 'source' worksheet is activated. It appears as follows:

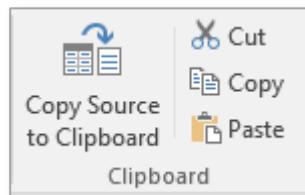


It contains the following major groups:

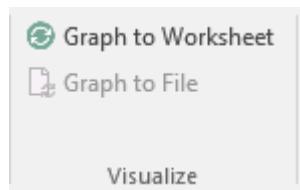
- [Source](#)
- [Clipboard](#)
- [Visualize](#)
- [Tools](#)
- [Web Resources](#)

Source

Label	Control Type	Description
Save Source	Button	Saves the Graphviz source code displayed on the 'source' worksheet to a file.
Create	Button	Creates the Graphviz source code from the information in the 'data' worksheet without invoking Graphviz to render a graph. This action is useful if you want to change the indentation of the subgraphs.
Reset	Button	Clears all data on the source worksheet, but leaves the headings.
Indent	Dropdown List	Number of spaces equaling a tab indentation

Clipboard

Label	Control Type	Description
Copy Source to Clipboard	Button	Selects all the Graphviz source code and copies it to the clipboard.
Cut	Button	Standard Excel 'Cut'
Copy	Button	Standard Excel 'Copy'
Paste	Button	Standard Excel 'Paste'

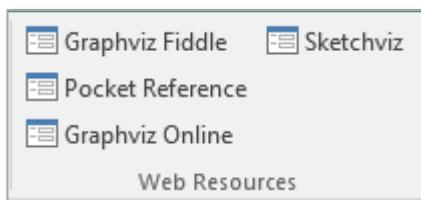
Visualize

Label	Control Type	Description
Graph to Worksheet	Button	<p>Graphs the Graphviz data on the source worksheet using the settings on the 'Graphviz' ribbon tab and displays the resulting graph on the 'graph' worksheet.</p> <p>If the graph fails to render you will receive a message saying it failed, but limitations of the code used to invoke Graphviz prevent returning the actual Graphviz error messages. To see those messages, you must use a different tool.</p> <p>Note also that the exchange of data is one direction. Changes made on the 'data' worksheet can be generated as 'source', but changes made on the 'source' worksheet will not be detected and fed back to the 'data' worksheet.</p>
Graph to File	Button	Graphs the Graphviz data on the source worksheet using the settings on the 'Graphviz' ribbon tab and writes the graph to a file. All the restrictions noted for the "Refresh Graph" button apply to this action as well.

Tools

Label	Control Type	Description
GVEdit	Button	<p>Launches the GVEdit tool which comes with Graphviz.</p> <p>Note: This tool may or may not be present in the latest edition of Graphviz for Windows. It is present in the older version 2.38 release so the button has been retained in anticipation that the program will be released again in a future Graphviz release.</p>

Web Resources



The 'Web Resources' group dynamically supports 1-6 buttons which can launch the user's default browser and display a web page. Button text, tool tips, and the URL are specified in the 'settings' worksheet. The worksheet as distributed contains three links to online Graphviz rendering tools. They are:

Label	Control Type	Description
GraphvizFiddle	Button	Graphviz Fiddle draws DOT language [directed] graphs. It is a dual-view Graphviz editor and playground for Graphviz written by Hermann Stamm-Wilbrandt.
Pocket Reference	Button	An on-line code editor for Graphviz written by Josh Hayes-Sheen for Computer Science students.
Graphviz Online	Button	Online Graphviz editor with syntax highlighting.
Sketchviz	Button	Online Graphviz editor which creates the graph as if it were sketched by hand.

The values associated with these buttons are located in the 'settings' worksheet in the 'Ribbon Options' location, in the 'Source Tab'

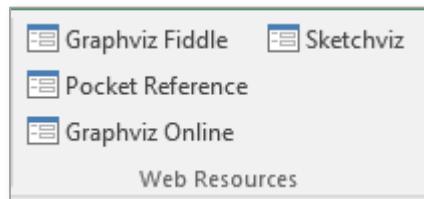
Ribbon Options

Source Tab

Button 1 Text:	Graphviz Fiddle	show
URL:	http://stamm-wilbrandt.de/GraphvizFiddle/	
Screentip:	On-line code editor for Graphviz	
Supertip:	GraphvizFiddle draws DOT language [directed] graphs. It is a dual-view Graphviz editor and playground for Graphviz written by Hermann Stamm-Wilbrandt.	
Button 2 Text:	Pocket Reference	show
URL:	http://graphviz.grevian.org/graph	
Screentip:	On-line code editor for Graphviz	
Supertip:	An on-line code editor for Graphviz written by Josh Hayes-Sheen for Computer Science students.	
Button 3 Text:	Graphviz Online	show
URL:	http://dreampuf.github.io/GraphvizOnline/	
Screentip:	On-line code editor for Graphviz	
Supertip:	Online Graphviz editor with syntax highlighting.	
Button 4 Text:	Sketchviz	show
URL:	https://sketchviz.com/new	
Screentip:	On-line code editor for Graphviz	
Supertip:	Renders Graphviz diagrams as if sketched by hand.	
Button 5 Text:		hide
URL:		
Screentip:		
Supertip:		
Button 6 Text:		hide
URL:		
Screentip:		
Supertip:		

Working Interactively with Graphviz over the Internet

There are public web sites which will allow you to edit DOT code in a browser window which you may use to edit DOT code interactively. You can launch these sites through the 'Web Resources' button on the 'Source' ribbon.



The sites appear as follows:

GraphvizFiddle

GraphvizFiddle draws DOT language [directed] graphs, see DOT User's Guide (on <http://graphviz.org>). It is an online code editor and playground for Graphviz dot layout tool, see [blog posting](#) for details.

```
strict graph "main"
{
    layout="dot";
    imagepath="C:\vsource;C:\xls2gv";
    rankdir="TB";

    "Parts" -> "Assembly";
    "Assembly" -> "Paint";
    "Paint" -> "Quality Control";
    "Quality Control" -> "Assembly";
    "Quality Control" -> "Paint";
    "Quality Control" -> "Shipping";
}
```

SVG output

GraphViz Pocket Reference

Make A Graph

Enter your DOT definition in the box below, and click Generate to display the resulting graph.

```
strict graph "main"
{
    layout="dot";
    imagepath="C:\vsource;C:\xls2gv";
    rankdir="TB";

    "Parts" -> "Assembly";
}
```

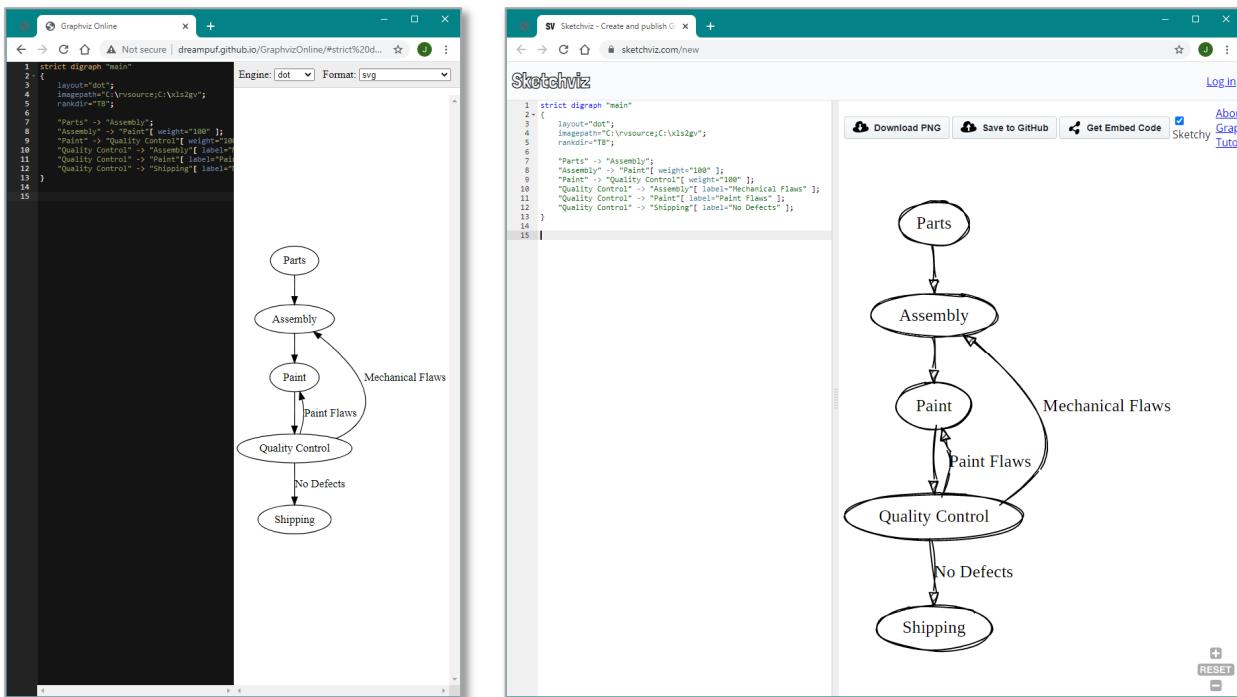
Layout Method (Just use trial and error to find the method you think works best)

Results

<http://stamm-wilbrandt.de/GraphvizFiddle/>

<http://graphs.grevian.org/graph>

Excel to Graphviz for Win OS



<http://dreampuf.github.io/GraphvizOnline/>

<https://sketchviz.com/new>

These sites work best with small graphs and cannot handle features like displaying images. There is no guarantee that these sites will continue to operate into the future. They do, however, make it easy to edit DOT graphs, quickly see the results, and learn the DOT programming language without having to install Graphviz.

WORKING WITH LARGE DATA SETS

It is very easy to collect large amounts of data in Excel, and Graphviz has powerful algorithms to determine how to place nodes and draw the relationships. You can reach a saturation point where the graph appears as a jumbled mess. This chapter will describe how you can extract insights from a large data set.

Large Data Set Example - 'The Beatles' Music Catalog

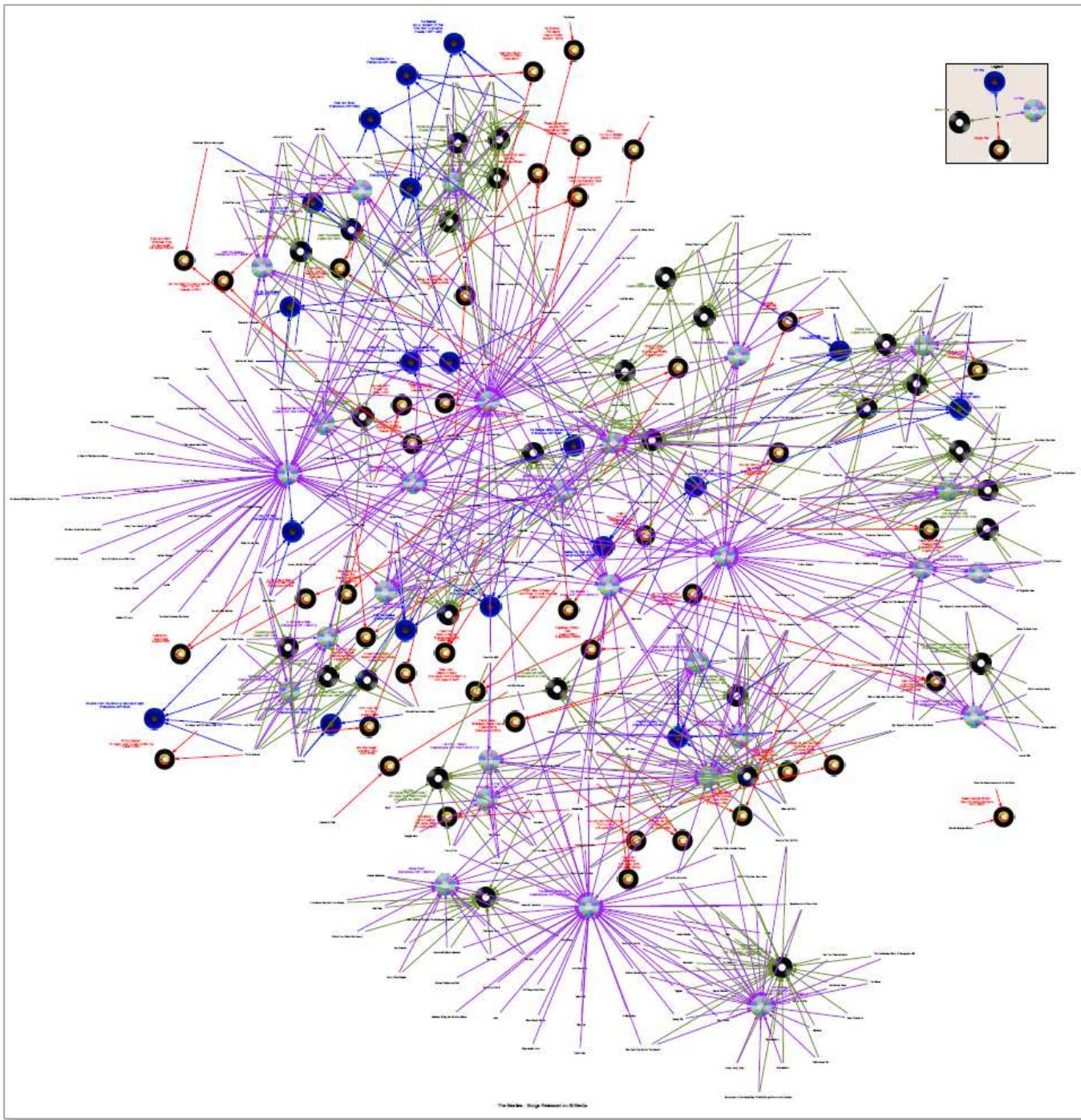
The Beatles were an English rock band, formed in Liverpool in 1960. With members John Lennon, Paul McCartney, George Harrison and Ringo Starr, they became widely regarded as the foremost and most influential act of the rock era.

The Beatles were very prolific in releasing music on various media such as 45 RPM records, LP records, EP records, and Compact Disc (CD) as technology changed over the years. In addition, songs were often regrouped into greatest hits compilations, or regional versions (US vs UK). Here are some statistics:

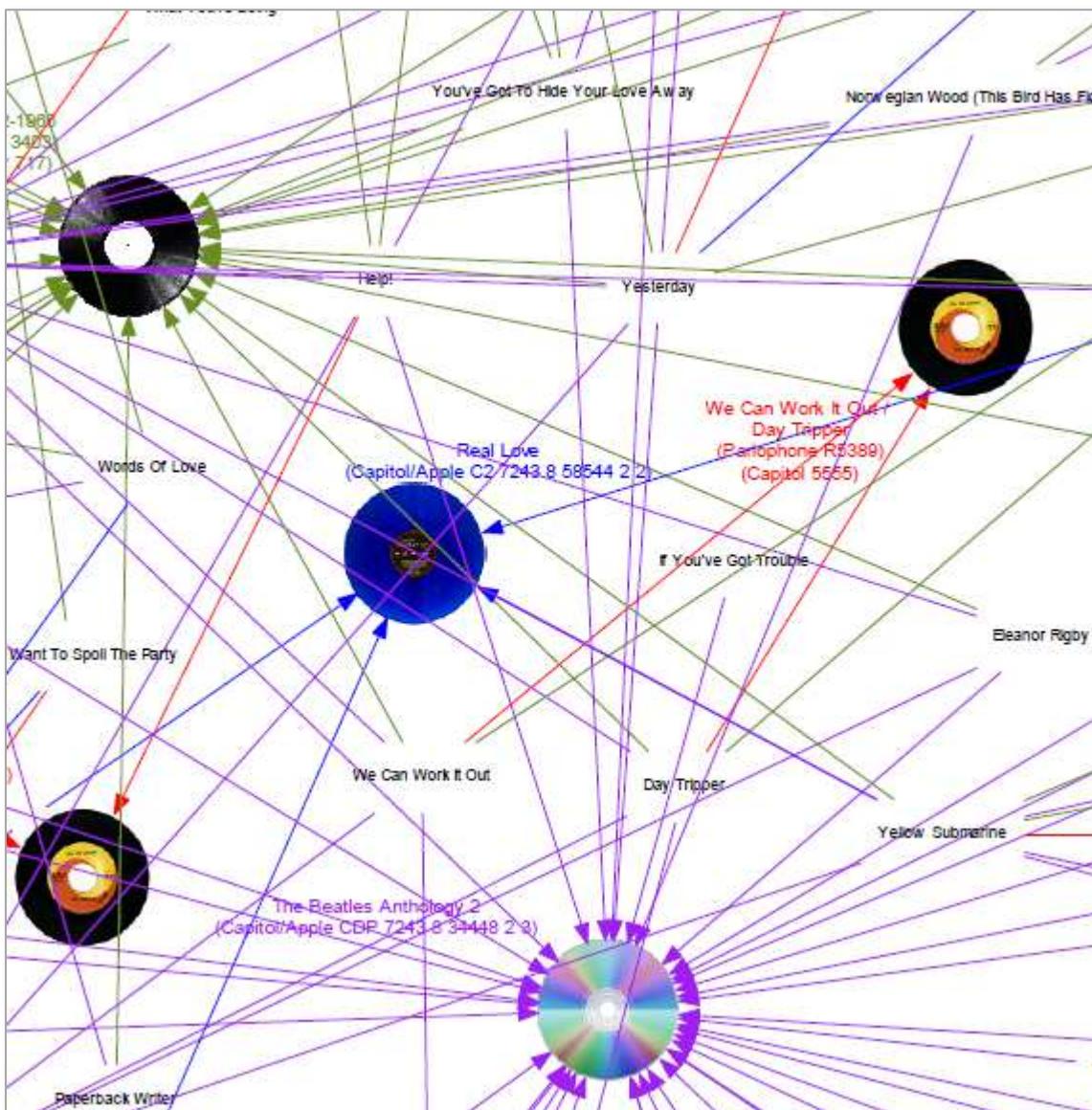
- 289 Songs released over 16 years
- 42 Singles (45 RPM) Released
 - 84 relationships between songs and 45 RPM singles
- 19 EP albums released
 - 78 relationships between songs and EP albums
- 29 LP albums (33 1/3 RPM) released
 - 391 relationships between songs and LP albums
- 28 CDs Released
 - 568 relationships between songs and Compact Discs

A Beatles Song Cross Reference at <http://www.beatlesagain.com/btsongs.html> was used to create a large data set within the Relationship Visualizer to give Graphviz a challenge. In most cases the spreadsheet will not be shown here, but it is provided in the "samples" directory of the Relationship Visualizer distribution.

Graphviz is up to the challenge of graphing the large data set, but due to the large number of relationships the graph comes out to be a bit of a jumbled mess, as shown below:



By creating the Graphviz output as a PDF file, text searches for songs such as "Yesterday" can be performed in Adobe Acrobat. Once the text is found once can zoom in to try to get a detailed view, as is shown below:



This just serves to confirm that you can't get much meaning out of this graph. Let's look at some ways we can work with the data to get at the information.

Use Views to Simplify the Graphs

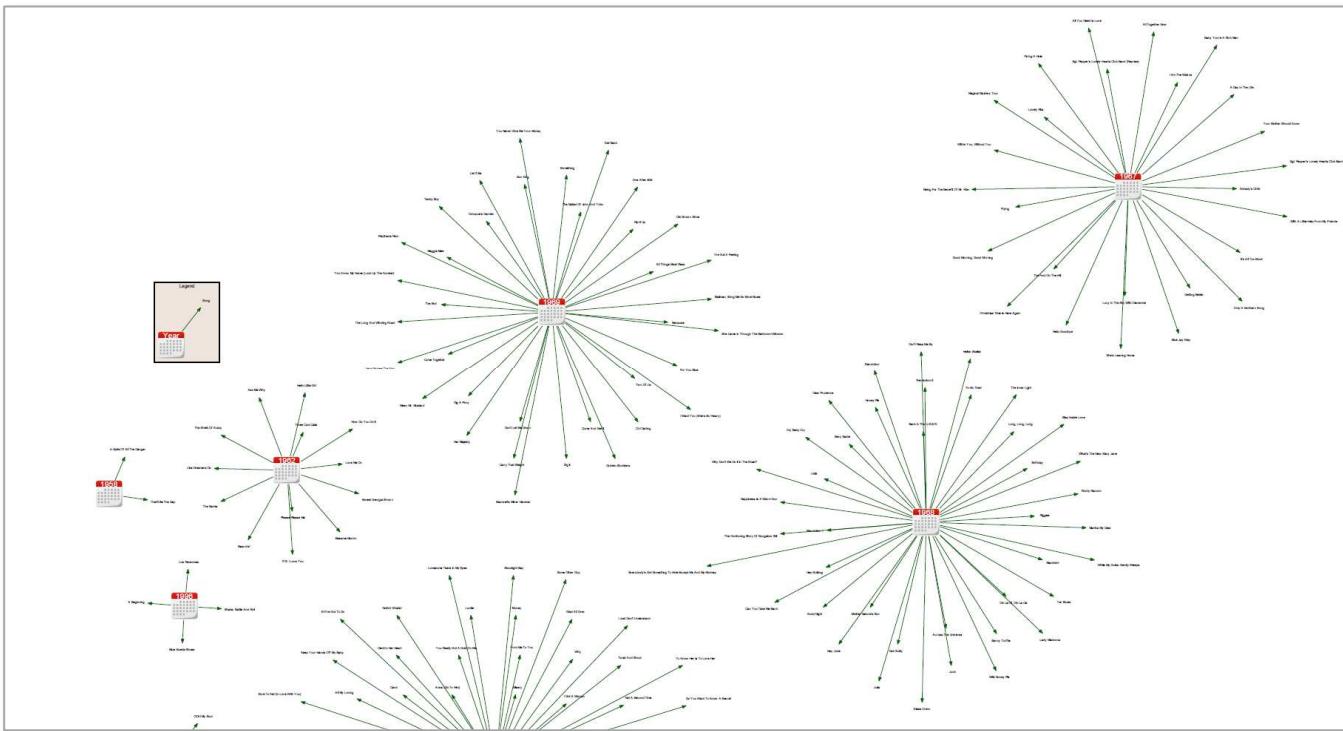
The first technique to simplify the graphs is to use views of the data. How to create views are described in [Creating Views](#) on page 145, and the lesson will not be repeated here. For the large data set, we created views by Year, Album, 45 RPM Single, CD, etc. which have 'yes' switches for the relevant media, and 'no' switches for the others.

The resulting 'styles' worksheet appears as:

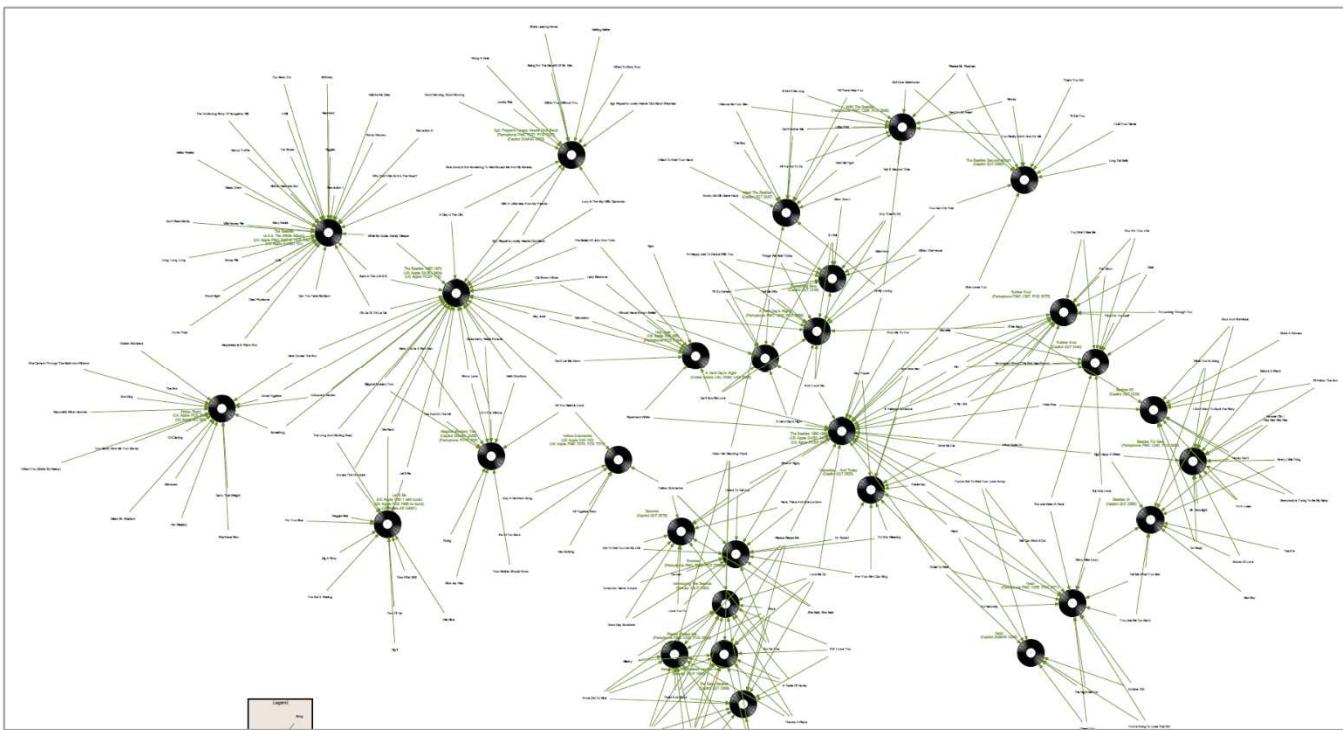
	A	B	C	D	E	F	G	H	I	J	K
	Style Name	Format	Object Type	Songs Released on All Media	Songs Released by Year	Songs Released on LP Album	Songs Released on 45 RPM Single	Songs Released on EP	Songs Released on Compact Disc	Songs Released on Single or LP Album	
1											
2	node		node	yes	yes	yes	yes	yes	yes	yes	
3	edge		edge	yes	yes	yes	yes	yes	yes	yes	
4	subgraph-open		subgraph-open	yes	yes	yes	yes	yes	yes	yes	
5	subgraph-close		subgraph-close	yes	yes	yes	yes	yes	yes	yes	
6	keyword		keyword	yes	yes	yes	yes	yes	yes	yes	
7	native		native	yes	yes	yes	yes	yes	yes	yes	
8	Song	shape="none", fontcolor="black", fontname="Arial", fontsize="8"	node	yes	yes	yes	yes	yes	yes	yes	
9	Album	shape="ellipse", color="none", fixedsize="True", fontcolor="olivedrab", fontname="Arial", fontsize="10", height="1.00", width="1.00", image="lp_record.png"	node	yes	no	yes	no	no	no	yes	
10	Single	shape="ellipse", color="none", fixedsize="True", fontcolor="red", fontname="Arial", fontsize="10", height="1.00", width="1.00", image="45_rpm_single.png"	node	yes	no	no	yes	no	no	yes	
	EP	shape="ellipse", color="none", fixedsize="True", fontcolor="blue", fontname="Arial", fontsize="10",	node	yes	no	no	no	yes	no	no	

We can then iterate though the columns on the 'settings' worksheet to create the various views.

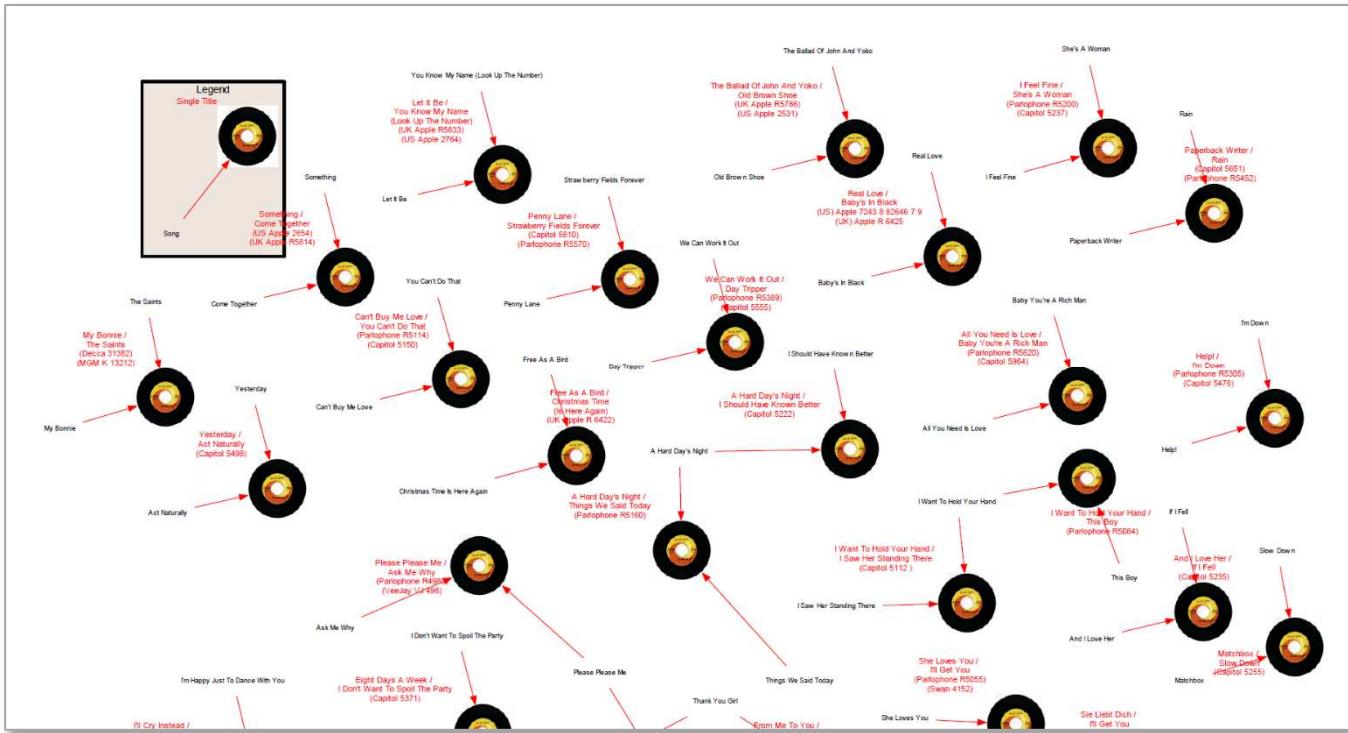
Songs Released by Year



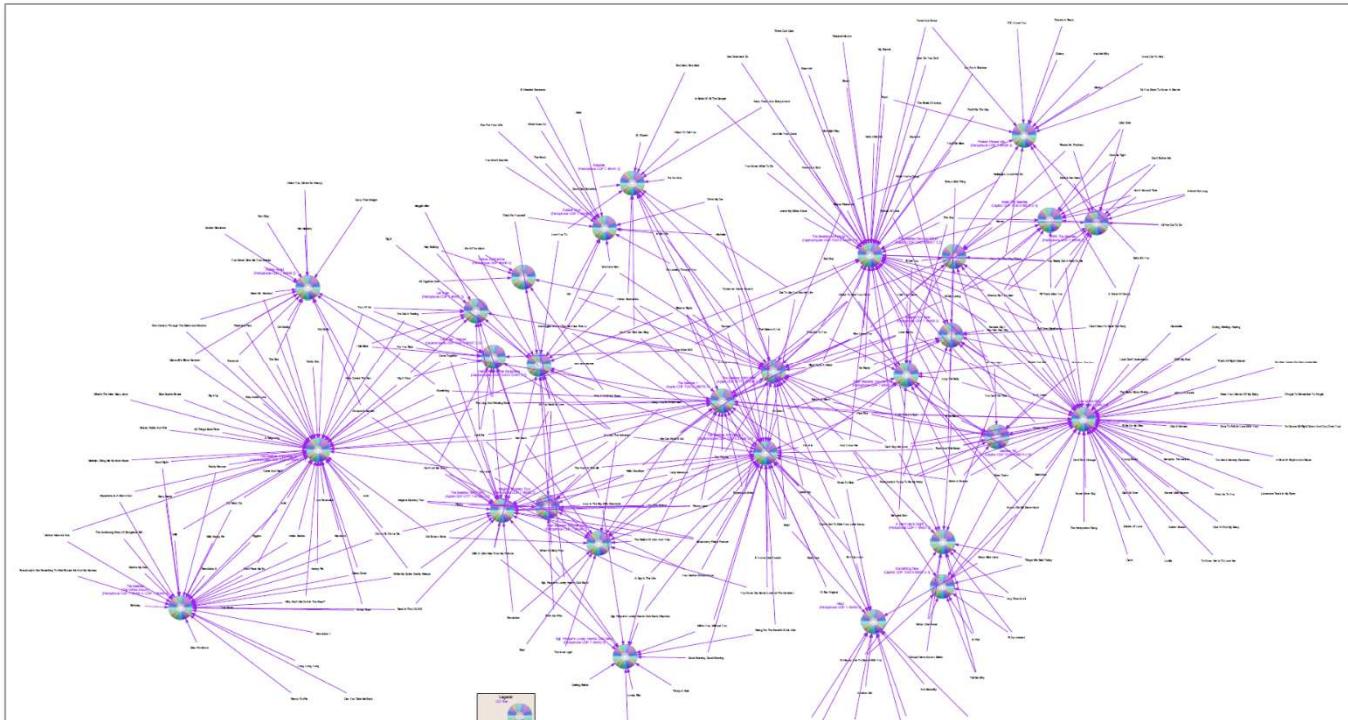
Songs Released on LP Album



Songs Released on 45 RPM Single

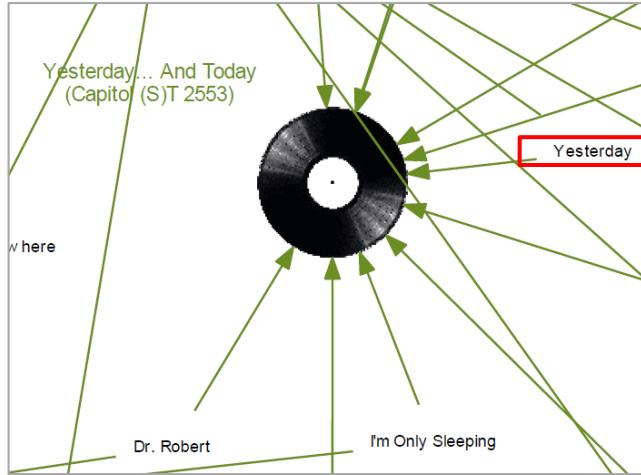


Songs Released on CD

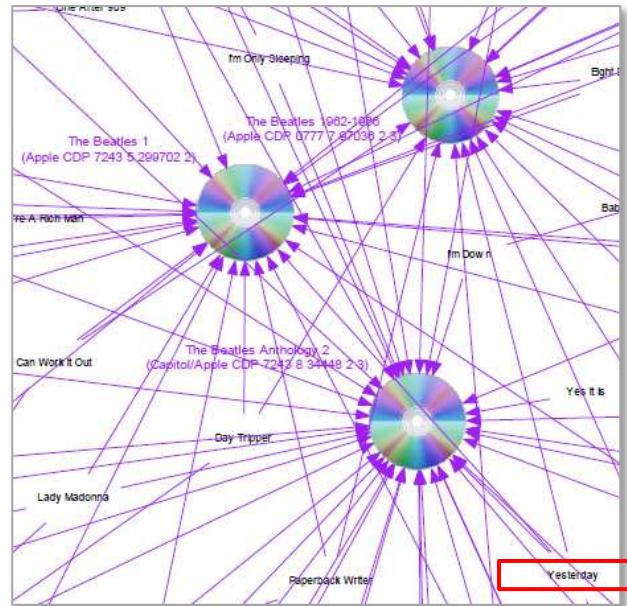


These views are useful for searching for patterns such as what were the most prolific years, or what songs were regrouped and released on multiple albums.

If we attempt to look at the song level within the views, it has become a little easier to find the information we are looking for. If we continue to look for the song "Yesterday" we see:



'Yesterday' shown as released on LP Album



'Yesterday' shown as released on CD

While this is a little more useful, it is still difficult to draw conclusions on individual songs. Let's see how we can create a view which shows just the song 'Yesterday' and all the media it was released on.

Use a Data Subset to Get to Answers

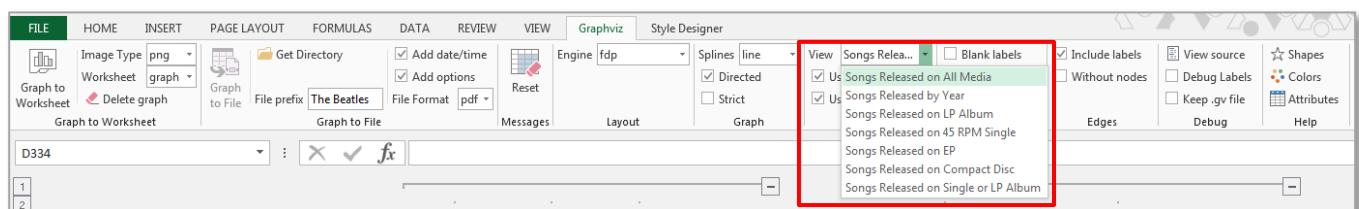
We can take advantage of the two features in the Relationship Visualizer. One is the ability to eliminate data from the data set without deleting the rows by commenting out records. The second is the feature to drop orphan relationships. If we comment out all the songs but "Yesterday" the Relationship Visualizer will discard all nodes (i.e., media) and relationships which are not associated with that song.

Commenting out the other songs with the "#" character turns the row green and makes the data appear as:

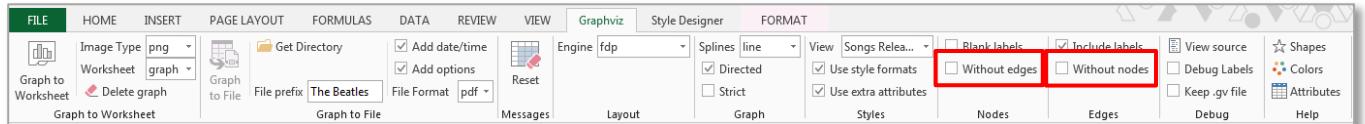
A	B	C	D	E	F	G	H
1	Item	Label	Outside Label	Tail Label	Head Label	Is Related To Item	Style Name
343 #	Yellow Submarine	Yellow Submarine					Song
344 #	Yer Blues	Yer Blues					Song
345 #	Yes It Is	Yes It Is					Song
346	Yesterday	Yesterday					Song
347 #	You Can't Do That	You Can't Do That					Song
#	You Know My Name (Look Up The Number)	You Know My Name (Look Up The Number)					Song
348							
349 #	You Know What To Do	You Know What To Do					Song
350 #	You Like Me Too Much	You Like Me Too Much					Song
#	You Never Give Me Your Money	You Never Give Me Your Money					Song
351							
#	You Really Got A Hold On Me	You Really Got A Hold On Me					Song
352							
353 #	You Won't See Me	You Won't See Me					Song
354 #	You'll Be Mine	You'll Be Mine					Song
#	You're Going To Lose That Girl	You're Going To Lose That Girl					Song
355							
#	You've Got To Hide Your Love Away	You've Got To Hide Your Love Away					Song
356							
357 #	Young Blood	Young Blood					Song
358 #	Your Mother Should Know	Your Mother Should Know					Song
#	Singles released by the Beatles						
359							
360	S1		My Bonnie / The Saints (Decca 31382)				Single

On the 'settings' worksheet we must make sure that:

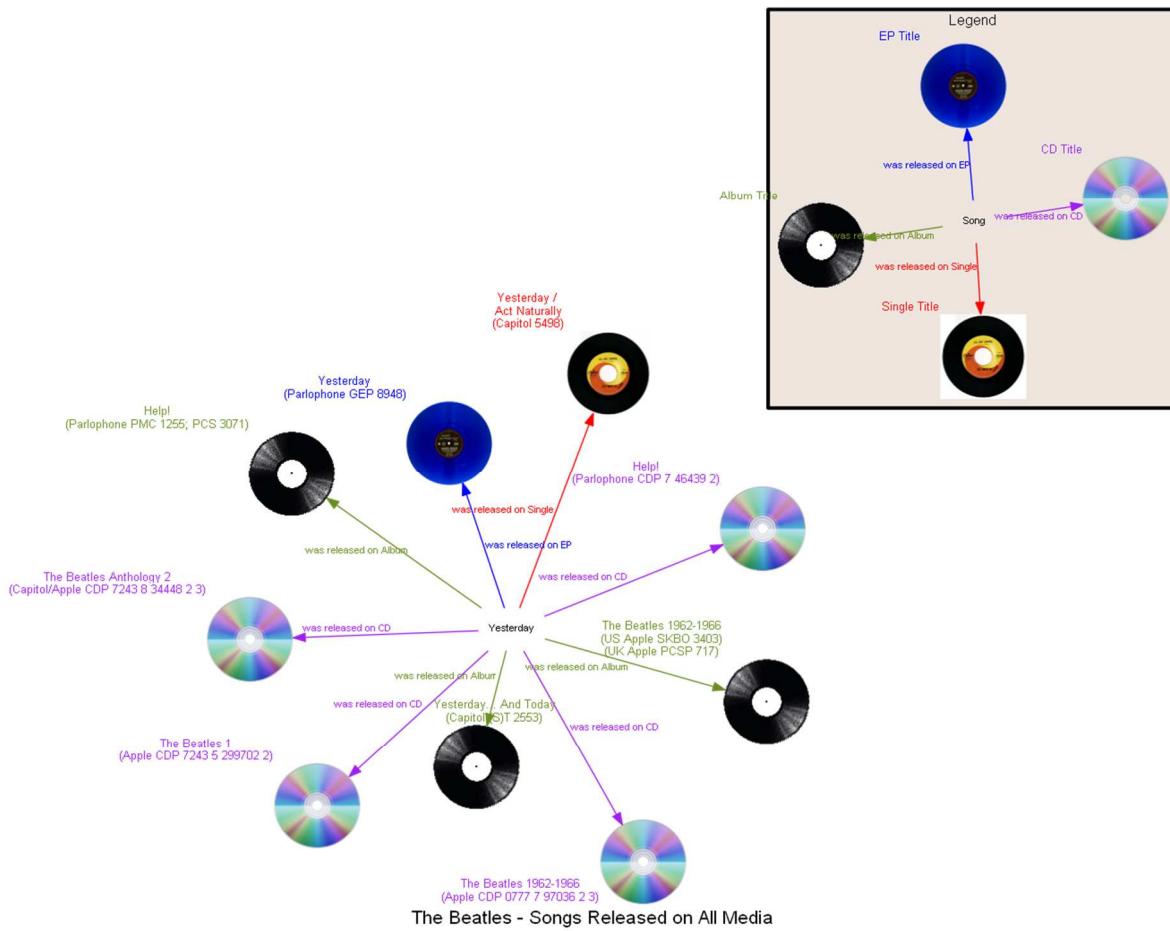
1. The 'styles' section of the 'Graphviz' ribbon tab has the View setting set to "Songs Released on All Media". This view will bring in all the various media types that the song "Yesterday" was released on.



2. The 'Nodes' section of the 'Graphviz' ribbon tab has "Without edges" unchecked, and the "Edges" section has "Without nodes" unchecked. These settings cause unreferenced nodes and edges to be filtered out of the graph.



When we press the "Refresh Graph" button we now get a simplified graph which looks as follows:



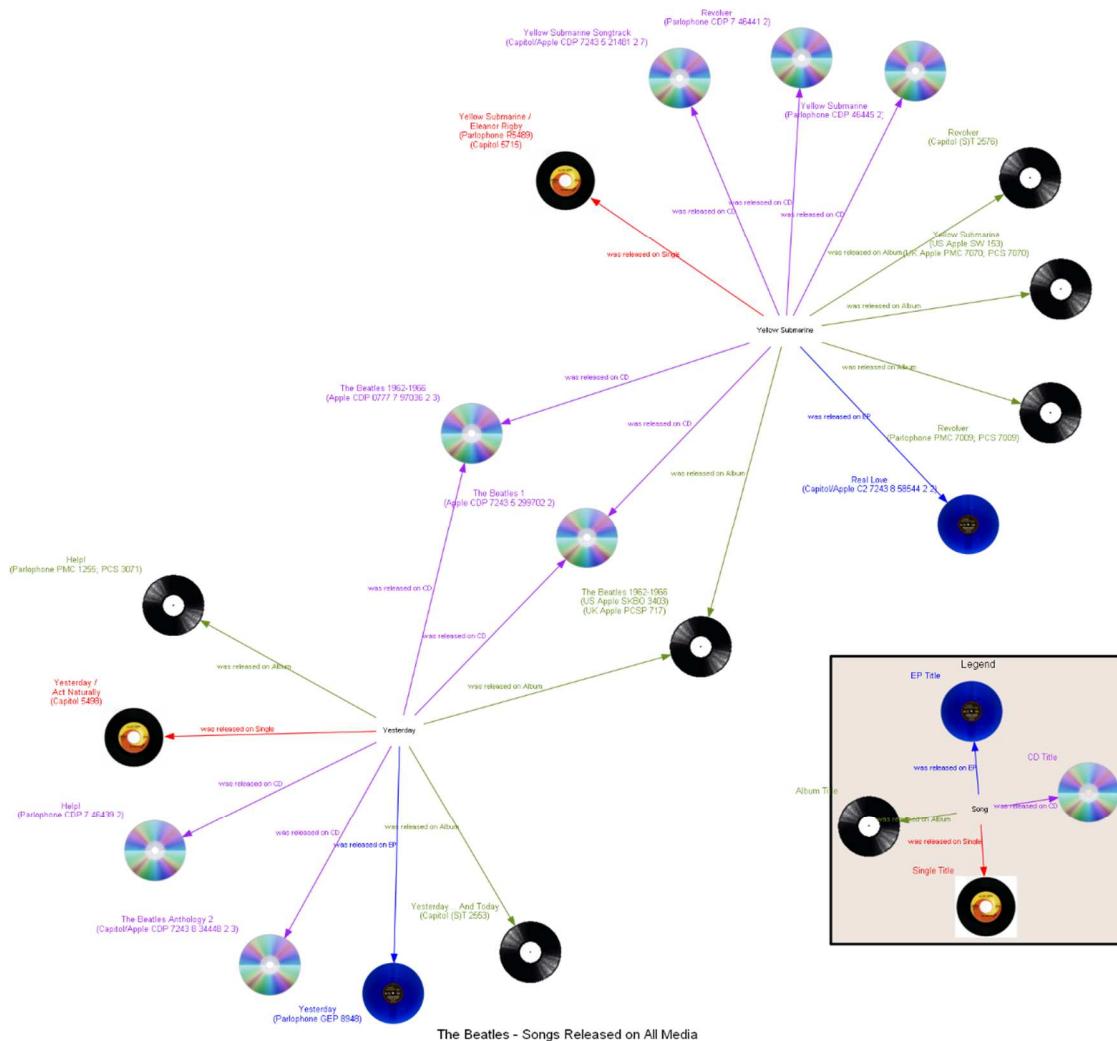
Now it becomes quickly and visually apparent that "Yesterday" was released on 4 CDs, 3 LP albums, a 45 RPM single, and an EP records. Each depiction provides the name of the media the song was released on.

Now assume that we are trying to solve a problem of what is the ideal media to buy to obtain the songs "Yesterday" and "Yellow Submarine" (chosen for this example just because both songs start with the letter Y). Return to the 'data' worksheet and remove the comment indicator '#' from the song "Yellow Submarine".

The data will now look as follows:

A	B	C	D	E	F	G	H
1	Item	Label	Outside Label	Tail Label	Head Label	Is Related To Item	Style Name
342 #	Words Of Love	Words Of Love					Song
343	Yellow Submarine	Yellow Submarine					Song
344 #	Yer Blues	Yer Blues					Song
345 #	Yes It Is	Yes It Is					Song
346	Yesterday	Yesterday					Song
347 #	You Can't Do That	You Can't Do That					Song

When we press the "Refresh Graph" button we get a graph which looks as follows:



The graph now depicts the additional media on which the song "Yellow Submarine" was released, but what we also see is that between the two songs there are some intersecting relationships. Both songs were released on 2 CDs ('The Beatles 1', and 'The Beatles 1962-1966') as well as an LP album ('The Beatles 1962-1966'). The most efficient way to obtain both songs is to buy one media from those three choices.

CHANGING MASTER SETTINGS

The 'settings' worksheet is where you can customize settings overarching settings which control how the graph is created, specify directories containing images to use in the graph, control how long the graphing engine is allowed to run, specify Graphviz command line parameters, and modify the order in which the worksheet columns are laid out. The following sections provide a brief description of the settings:

Graph Options

Command Options	'Graphviz' Tab	'Source' Tab	'data' Worksheet	'sql' Worksheet
Graph Options	'Extensions' Tab	'Exchange' Tab	'source' Worksheet	'styles' Worksheet
<p><i>Image Path:</i> <input type="text"/> ...</p> <p><i>Additional Graph Options:</i> <input type="text"/></p> <p><i>"Graph to Worksheet" Picture Name:</i> <input type="text"/></p>				

These settings control the graph options used to produce the diagrams.

Setting	Description
Image Path	<p>Specifies one or more directories when images to include in the graph are stored. The Relationship Visualizer looks for images in the same directory where the spreadsheet is stored first, followed by the directory specified here. The folder picker allows you to select one directory, however multiple directories can be specified if the directories are separated by a semi-colon ";".</p> <p>If you use the 'style designer' worksheet to design an image node the directory containing the image will be automatically appended to the value here, unless the directory is already listed or is the directory where the Excel spreadsheet is stored.</p>
Additional Graph Options	<p>A means to insert additional Graphviz graph directives into the graph at the top-most level. To use this setting, you must understand how Graphviz works. This setting can be used to control aspects such as node spacing. For example:</p> <pre>sep="+30,30"</pre>
"Graph to Worksheet" Picture Name	The name to assign to the bitmap which is created by the "Refresh Graph" button on the "Data" worksheet. If blank, Excel assigns a name.

Command Options

The 'Command Line Options' settings allow you to pass parameters to the Graphviz command line programs.

Command Options	'Graphviz' Tab	'Source' Tab	'data' Worksheet	'sql' Worksheet
Graph Options	'Extensions' Tab	'Exchange' Tab	'source' Worksheet	'styles' Worksheet
<i>Additional CMD Parameters:</i> <input type="text"/> <i>Cancel graphing if not complete in:</i> <input type="text" value="60"/> seconds.				
<i>Path to dot.exe:</i> <input type="text"/>				

Setting	Description
Additional CMD Parameters	A means to pass additional command line parameters to the layout engine program. For example: -Efontsize="8" -Efontname="Arial" would override the default edge format. Setting the font size to 8 points, and the font to Arial for all edges which do not have explicit style specifications.
Cancel graphing if not complete in _____ seconds	The maximum number of seconds the graphing engine can run. If set to 0, no time limit exists.
Path to dot.exe	The path to the dot executable if Graphviz is not installed/not available via the PATH environment variable.

'data' Worksheet

These settings allow you to rearrange the column layout of the 'data' worksheet, and restrict which rows get processed.

Command Options	'Graphviz' Tab	'Source' Tab	'data' Worksheet	'sql' Worksheet																																	
Graph Options	'Extensions' Tab	'Exchange' Tab	'source' Worksheet	'styles' Worksheet																																	
<p><i>Comment/Error Flag Column:</i></p> <p>"Item" Column:</p> <p>"Tail Label" Column:</p> <p>"Label" Column:</p> <p>"External Label" Column:</p> <p>"Head Label" Column:</p> <p>"Related Item" Column:</p> <p>"Style" Column:</p> <p>"Attributes" Column:</p> <p>"Messages" Column:</p> <p>"Graph" Column:</p>	<table border="1"> <tr><td>A</td><td>▼</td><td>show</td></tr> <tr><td>B</td><td></td><td>show</td></tr> <tr><td>C</td><td></td><td>hide</td></tr> <tr><td>D</td><td></td><td>show</td></tr> <tr><td>E</td><td></td><td>hide</td></tr> <tr><td>F</td><td></td><td>hide</td></tr> <tr><td>G</td><td></td><td>show</td></tr> <tr><td>H</td><td></td><td>show</td></tr> <tr><td>I</td><td></td><td>show</td></tr> <tr><td>J</td><td></td><td>hide</td></tr> <tr><td>K</td><td></td><td></td></tr> </table>	A	▼	show	B		show	C		hide	D		show	E		hide	F		hide	G		show	H		show	I		show	J		hide	K				<p><i>Heading Row:</i> <input type="text" value="1"/></p> <p><i>First Row:</i> <input type="text" value="2"/></p> <p><i>Last Row:</i> <input type="text" value="0"/></p>	
A	▼	show																																			
B		show																																			
C		hide																																			
D		show																																			
E		hide																																			
F		hide																																			
G		show																																			
H		show																																			
I		show																																			
J		hide																																			
K																																					

Setting	Description
Comment/Error Flag Column	The column used to place '#' comment characters which indicate that the rows should be treated as a comment and not be included in the data to be graphed. This column also serves a second purpose when errors are found in the data. An exclamation point "!" character is placed in the column to signal an error, and direct you to look in the Messages column for the cause of the error.
"Item" Column	The column within the 'data' worksheet where the "Item" values are located.
"Label" Column	The column within the 'data' worksheet where the "Label" values are located.
"Outside Label" Column	The column within the 'data' worksheet where the "Outside Label" values are located.
"Tail Label" Column	The column within the 'data' worksheet where the "Tail Label" values are located.
"Head Label" Column	The column within the 'data' worksheet where the "Head Label" values are located.
"Related Item" Column	The within the 'data' worksheet column where the "Related Item" values will be found

Setting	Description
"Style" Column	The Column within the 'data' worksheet where the "Style" name values are located.
"Attributes" Column	The column within the 'data' worksheet where the "Attributes" style attributes are located.
"Messages" Column	The column located within the 'data' worksheet where error messages are to be written to.
Heading Row	The row containing the column headings
First Row	First row of data.
Last Row	Last row of data definitions. When set to 0 all rows after the First Row containing data are used. First Row and Last Row are useful during graph development for specifying a subset of data to process when the overall amount of data is large.

'source' Worksheet

These settings allow you to rearrange the column layout of the 'source' worksheet.

Command Options	'Graphviz' Tab	'Source' Tab	'data' Worksheet	'sql' Worksheet
Graph Options	'Extensions' Tab	'Exchange' Tab	'source' Worksheet	'styles' Worksheet
	<i>Line Number Column:</i> <input type="button" value="A"/> <input type="button" value="B"/> <i>Graphviz Source Column:</i> <input type="button" value="A"/> <input type="button" value="B"/>		<i>Heading Row:</i> <input type="button" value="1"/> <i>First Row:</i> <input type="button" value="2"/> <i>Tab Indent Spaces:</i> <input type="button" value="4"/>	

Setting	Description
Line Number Column	The column which contains the line number of the Graphviz DOT source code.
Graphviz Source Column	The column where the Graphviz DOT source code gets placed
Heading Row	The row containing the column headings
First Row	First row of data.
Tab Indent Spaces	The number of blanks to insert based upon the depth of subgraphs within the graph.

'sql' Worksheet

These settings allow you to rearrange the column layout of the 'sql' worksheet, and restrict which rows get processed.

Command Options	'Graphviz' Tab	'Source' Tab	'data' Worksheet	'sql' Worksheet
Graph Options	'Extensions' Tab	'Exchange' Tab	'source' Worksheet	'styles' Worksheet
<i>Comment/Error Flag Column:</i> A <i>SQL Statement Column:</i> <i>Excel File Column:</i> <i>Status Column:</i> D		<i>Heading Row:</i> 1 <i>First Row:</i> 2		

Setting	Description
Comment/Error Flag Column	The column used to place '#' comment characters which indicate that the rows should be treated as a comment and not be included in the set of SQL statements which get executed.
SQL Statement Column	The column containing the SQL SELECT statements to execute
Excel File Column	The column containing the paths to the Excel files containing the data which the SQL SELECT statements should be executed against.
Status Column	The column where the status (Success/Failure/Skipped) of the SQL SELECT statement, along with any error messages is written to.
Heading Row	The row containing the column headings
First Row	First row of SQL SELECT statements to execute.

'styles' Worksheet

The "styles' Worksheet' settings allow you to rearrange the column layout of the 'styles' worksheet, and specify the range of rows to be included in the style collection at run-time.

Command Options	'Graphviz' Tab	'Source' Tab	'data' Worksheet	'sql' Worksheet
Graph Options	'Extensions' Tab	'Exchange' Tab	'source' Worksheet	'styles' Worksheet
<i>Comment Column:</i> <input type="button" value="A"/> <i>Style Column:</i> <input type="button" value="B"/> <i>Format Column:</i> <input type="button" value="C"/> <i>Object-Type Column:</i> <input type="button" value="D"/> <i>First Yes/No View Column:</i> <input type="button" value="E"/>		<i>Heading Row:</i> <input type="button" value="1"/> <i>First Row:</i> <input type="button" value="2"/> <i>Last Row:</i> <input type="button" value="0"/>		

Setting	Description
Comment Column	The column used to place '#' comment characters which indicate that the rows should be treated as a comment and not be included in the set of styles.
Style Column	The column where the name of the style is specified
Format Column	The column where the node, edge, or cluster format string is specified.
Object-Type Column	The column where the object-type is specified. The object-type is used to classify the style as a node, edge, subgraph-open, subgraph-close, keyword, or native type.
First Yes/No View Column	The first column containing Yes/No switches which control which styles will comprise the view.
Heading Row	The row which contains the column headings. The heading value of the "Yes/No Switch Column" is included in the graph's file name after the file name prefix.
First Row	First row of style definitions (i.e., where the definitions begin)
Last Row	Last row of style definitions. When set to 0 all rows after the First Row containing data are used.