Inkscape - Part 136

Inkscape development has certainly picked up in pace over the past few years, which is great for users… but not so good for volunteer writers trying to extensively document all the features of the application in a monthly column. So I have slightly mixed feelings about the recent release of version 1.3. On the one hand it comes with various fixes and improvements, and some really impressive new features. On the other hand, I have barely started documenting the new features in version 1.2, so my backlog of topics has now grown many times longer!

In an effort to catch up with the Inkscape developers, I'm going to use this article to briefly revisit the 1.2.x features I've covered and describe any changes to them in 1.3. Then from next month I'll be describing the changes and additions that have taken place in either 1.2.x or 1.3, but documenting them as they currently look and behave in the latest version.

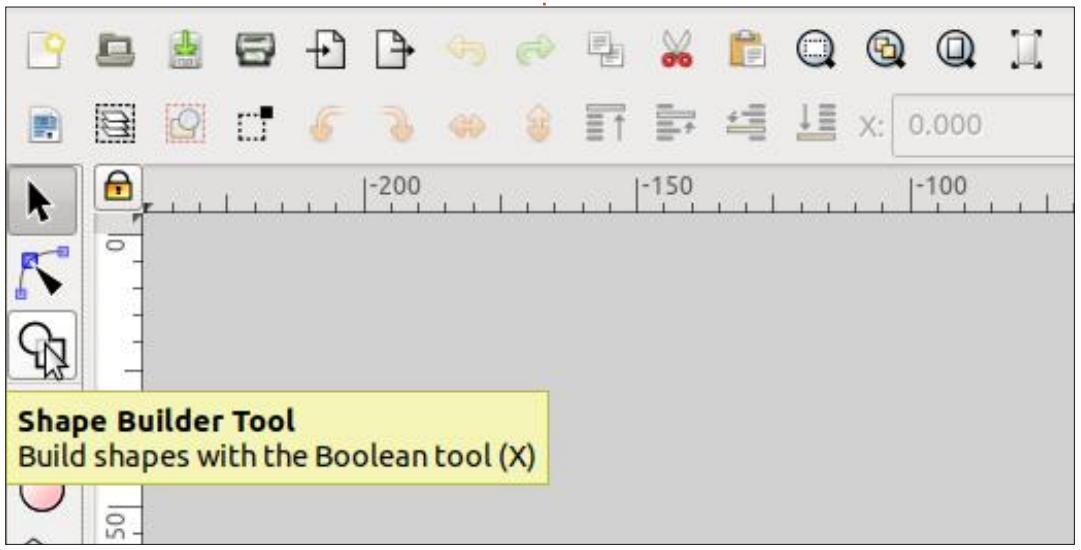
I strongly recommend all Inkscape users to download the 1.3 release. The best way to do this right now is probably to visit the project's website at https://inkscape.org and just click on the 'Download Now!' button. Linux users have the option of downloading an AppImage file, or using a PPA. I'm using the AppImage, simply because I need to have multiple different versions available while writing this column, which is not so straightforward with a PPA.

Now, let's revisit the 1.2 features that I covered in the previous few articles…

Icons and the toolbar

In FCM #188 I introduced some of the changes that had taken place with the main toolbar, and with the icons within the application. I also drew attention to the fact that (on my machine, at least) the symbolic icons still seemed to creep into parts of the UI, even though I had chosen a full color theme.

With 1.3 there have been few changes to the toolbar. It can still be resized, and individual tools can be shown or hidden via the Preferences dialog, as I described previously. In fact, the only significant change to the toolbar is the addition of another new tool – the Shape Builder – which will be the subject of a future article.



I'm also pleased to note that the symbolic icons have disappeared, and the icons in my UI all come from the high color icon set that I prefer (being a dyed-in-the-wool Inkscape user of old). Interestingly my 1.2.x versions also now show the correct icons. Perhaps running 1.3 has updated a common system file, fixing the problem more generally.

Snap controls

Issue #188 was also where I introduced the new Snap Controls popup at the top-right of the Inkscape window. I covered both the Simple and Advanced modes that are available in this UI.

Version 1.3 fails to fix my biggest gripe with this popup – namely that switching to Simple mode then back to Advanced

resets all the many checkboxes to their defaults. I still feel that using a more hierarchical tree structure so that individual sections can be collapsed would be preferable to a forced choice between a minimalistic Simple mode that doesn't adequately describe its settings, and a comprehensive Advanced mode that can be overwhelming.

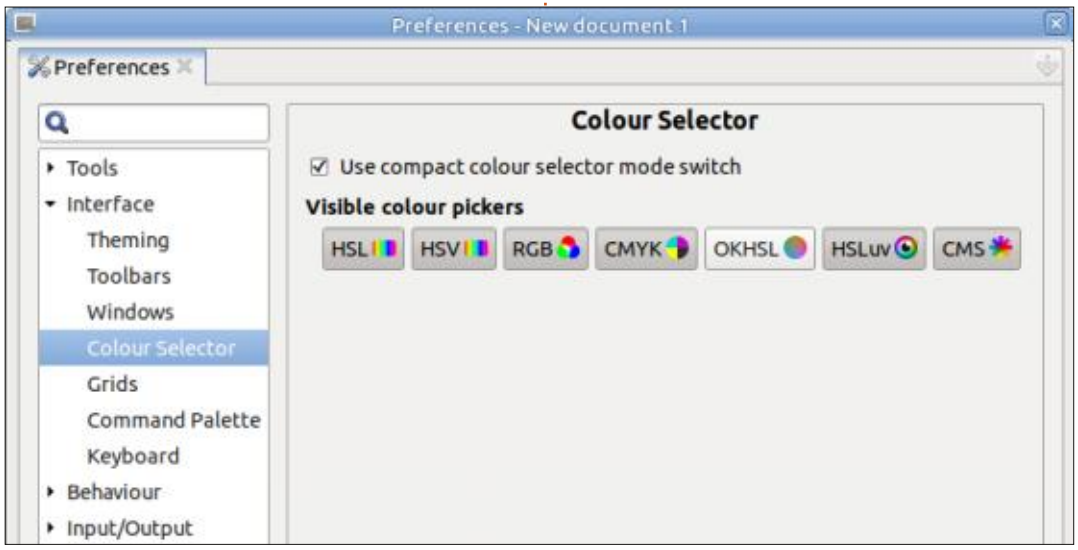
Perhaps not for the same reasons, but it seems that enough other users aren't happy with the new UI that the old Snap toolbar has been brought out of retirement for version 1.3. The option to switch to this mode isn't available from the Snap popup, so isn't as discoverable as it could be. Instead it's hidden away in the Interface > Toolbars section of the Preferences dialog. Here you can select between Simple, Advanced or Permanent, corresponding to the two views of the popup, or the traditional toolbar.

It's worth pointing out, however, that even with Permanent selected things aren't quite the same as older versions of Inkscape. The program used to have a badly named 'Custom' view, which offered absolutely no customisation. One thing it did do, however, was to move the Snap toolbar from the right of the window to the top. When enabling the Permanent option in 1.3 you get the Snap toolbar on the right, but as the Custom view no longer exists, and the toolbar can't be dragged to another place, that location is where it must remain. So if you're in that small crescent of the Venn diagram of users that both prefer a snap toolbar, and want it at the top of the window, you're still out of luck.

Color palette

In FCM #189 the subject of this column was colour selection. Specifically I introduced the new colour picker options in the Fill & Stroke dialog. Nothing much has changed in the dialog itself for 1.3, but there have been some changes in the Inkscape preferences related to the color pickers.

If you open Edit > Preferences then expand the Interface section, you'll find a new pane named 'Color Selector' (this screenshot is from my British English installation, so uses the UK spelling of 'Colour').



The first control here isn't new, but it has moved. I discussed it back in issue #190, when it lived in the 'Interface' pane directly. Basically it lets you switch between the old-but-classic method of changing color selectors using some tab-like buttons, or the new-and-compact approach of a pop-up menu. I prefer the former, as it's more immediate and requires fewer mouse clicks, but if you rarely switch between selectors then you might prefer something that takes up less space in the UI.

The second control consists of a set of toggle buttons that let you turn the individual color pickers on and off. If you never work with a color management system, for example (that would be most Inkscape users, I suspect) then you may wish to disable the 'CMS' picker. And perhaps you've read enough of these articles over the years to know that, internally, Inkscape only supports RGB colors, so the 'CMYK' picker is really a bit of a lie that may not be terribly useful to you. Whichever combination of pickers you enable will apply whether you use the compact or traditional mode switch.

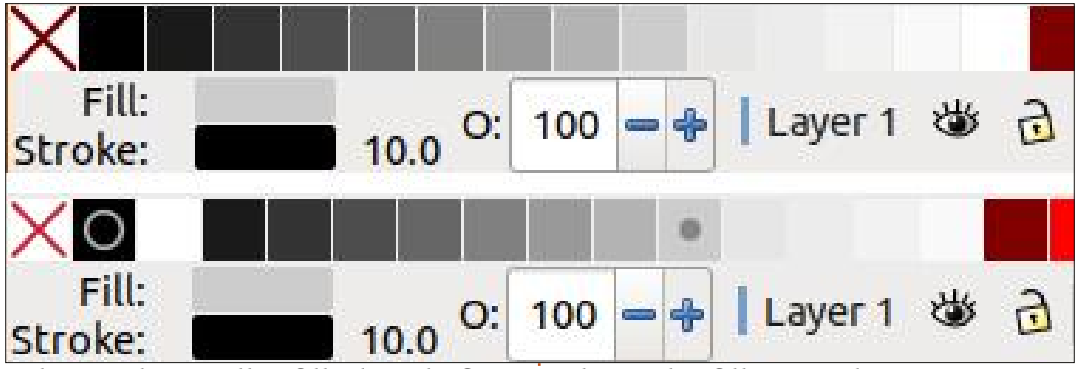
Version 1.3 adds a new picker to the options in the form of OKHSL. I'll let you look up the details of that color model yourselves, but I think it's worth enabling for most users. OKHSL has been specifically designed to try to maintain the perceptual brightness and saturation of colours when only the

hue changes. Using this can make for more consistent gradients, for example, and may make it easier to pick different colors which have similar intensity.

OKHSL has now made it into CSS – and, by extension, is valid in SVG. It is also rapidly being adopted by browsers. Because Inkscape still uses 8-bit RGB values internally, much like the CMYK picker this is a bit of a lie – though this one is perhaps more useful to everyday users. Unfortunately, extending Inkscape's code to be able to support true OKHSL values (and other color models now supported by CSS) is likely to be a huge undertaking, so I don't expect that to happen any time soon.

In issue #189 I also talked about the changes to the palette, at the bottom of the main Inkscape window. Little has changed here, but version 1.3 does introduce a small usability enhancement which really does make a big difference. Compare these screenshots – from 1.2.2 and 1.3 – for an object with a black stroke and gray fill.

Notice that 1.3 shows an open circle directly in the palette swatch for the currently selected stroke



color, and a smaller filled circle for the fill. These have been perfectly sized so that they are both clearly visible even when the fill and the stroke have both been set to the same color.

A subtle aspect of this feature is that the color used for each of the circles changes to contrast with the swatch color, ensuring that they never disappear from view entirely. Even more impressively, these indicators just do the right thing if you have multiple objects selected. If all the selected objects share a common fill or stroke color the relevant circle(s) will be visible. But where the fill or stroke are not common across all the selected objects, the corresponding circle won't be drawn, avoiding any confusion that might occur if colors are similar but not quite the same.

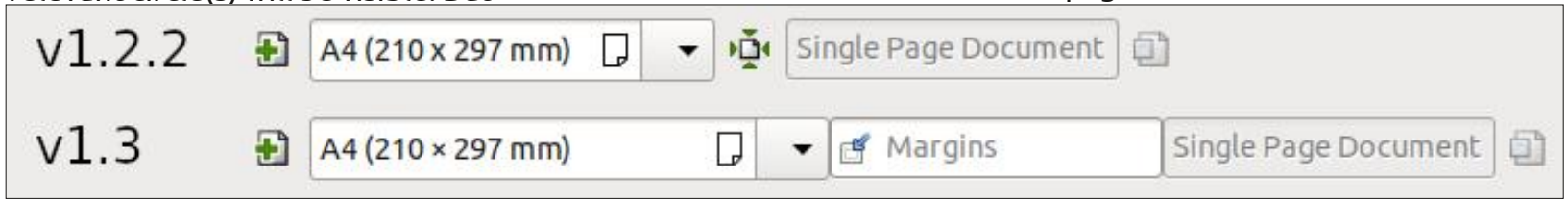
Multi-page documents

The multi-page tool was added in version 1.2 and I covered it in issue #192. When first clicking the tool, the UI has changed a little:

The resize button (the one with four arrows) has been removed, which I find to be a rather odd choice. Previously this would either resize the page to fit the contents (if you had nothing selected), or to fit the selected objects. You can still achieve the same effect using the equivalent button in the Document Properties dialog – but that only works for the first page of the document, not the currently selected page. As far as I can tell, there's no way to auto-size an arbitrary page to fit its content, which feels like a significant omission – or rather, removal – to me.

If you do wish to use the resize button in the Document Properties dialog, a new addition is that right-clicking away from any objects on the canvas will open a cut-down context menu which now offers a shortcut to open the dialog.

One thing I didn't like with the Page Tool's resize button in 1.2.x was that there was no way to include a little bit of padding or spacing around the objects. If you used the button the resultant page would hug your content closely. In most cases I prefer a little breathing space between my content and the very edge of the page, but neither the button in the page tool nor the one in Document



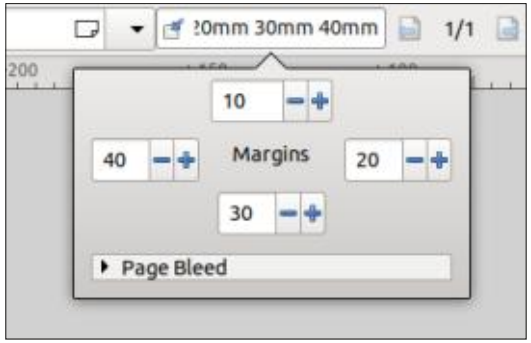
Properties offer that facility. So when I saw the new 'Margins' field in 1.3 I thought that, perhaps, that was the answer. Type a value in there, and the page would resize to suit the content (or selection), plus a bit extra for the margin. Alas, that's not what it does at all.

If you type a number in that field and press Enter (tabbing out of the field doesn't cut it) then Inkscape will draw a rectangle, offset by the specified amount within the page. You can think of it as being a little like a guideline, but rectangular – though it doesn't toggle on and off when you show or hide guidelines. It can, however, be used as a snap target, so if you know you need to allow, say, 5mm at the edge of your design for your particular printer's page handling, this can help you to avoid going outside the safe area. But it's only a guideline, and doesn't get factored in when telling Inkscape to resize the page.

Usually typing a single number in this box will suffice, but there may be times when you need different margins for each side of your page. For example, it's common to require a larger margin on the inside edge of the page to allow for binding if you are producing some artwork that will be made into a booklet. You can enter several space-separated numbers into this field, in a manner that might be familiar to web developers from the similar shortcuts CSS allows.

1 value: This will be used for all sides
2 values: The first dictates the top and bottom margins, the second sets the left and right margins
3 values: The first defines the top margin, the second is used for the left and right, and the third is used for the bottom
4 values: Top, right, bottom, left
5 or more values: All but the first four are deleted

You may notice that the margin box on the canvas also has some circular handles at the center of each side (only shown when the Page tool is active). You can drag these to dynamically adjust the values. Alternatively, clicking on the icon at the left of the field opens a small 'doorhanger' dialog in which you can set these values:



You'll notice there's also a section for 'Page Bleed' at the bottom of that dialog. Expanding this presents a single field for setting a bleed value that is used for all sides of the document. Again, you have to press Enter (or Return) to apply this value – tabbing won't do – and again this draws a guideline-like box which can be snapped to. In this case the box lives outside the page boundary.

You may not be familiar with the idea of 'bleed' in printing. It doesn't matter whether we're talking about a cheap inkjet in your house, or a huge industrial press that fills a warehouse, all printers are obviously mechanical devices. As such they have tolerances between components, parts can wear, and things can simply be misaligned from their perfect positions. If you want to print a solid background color in your design, therefore, keeping within the page boundaries could lead to a slightly offset print in which the white paper shows along one or two edges.

To compensate for this it's common to create your design such that it 'bleeds' off the edge of the page – that is, backgrounds and other objects extend a small way beyond the page boundary so that even if the paper is a little misaligned there will still be content to print which will cover-up the problem. For commercial printing a bleed of 5-10mm all round is common.

Conversely, margins are often specified to ensure that your content is safe from being trimmed when the final print is guillotined, for example.

So with these fields, Inkscape makes it a little easier for you to create designs to meet common commercial requirements. Set them to the values provided by the printing firm, and design accordingly: vital parts of the design should all be within the margins, while solid backgrounds or objects that must run right to the edge of the page should be

drawn up to the bleed line or beyond.

On the canvas, both the margin and bleed boxes on are extremely faint – so much so that I haven't bothered providing a screenshot, as I doubt they'd be visible in the magazine! Currently there seems to be no way to change their colors, which may cause a problems for some users. When treated purely as snap targets that may not matter as much, but if you wanted to use them as general guides just to get a visual idea of the placement of your objects, they may not be terribly useful for some users, or with some designs.

On the subject of snapping, both these boxes are enabled as snap targets when the 'Bounding Boxes' option is enabled in the Simple snap menu. In the Advanced menu they're handled by a single 'Page margins' option at the bottom – though when I tested this it was a little buggy, and I ended up having to switch back to Simple mode to get them to work again. With the Permanent snap toolbar there doesn't seem to be a way to toggle them at all!

None of the modes let you enable snapping for margins but not bleed, or vice versa. When designing for print I often treat the margins as a 'hard' edge that I can't exceed, whereas bleed is more of an advisory that it's okay to go beyond. Therefore I'd rather be able to enable these separately, but it's not a huge inconvenience either.

Finally, I can confirm that the JavaScript code I presented last month – for adding named views for multi-page documents so they can be easily viewed in a web browser – works perfectly in version 1.3.

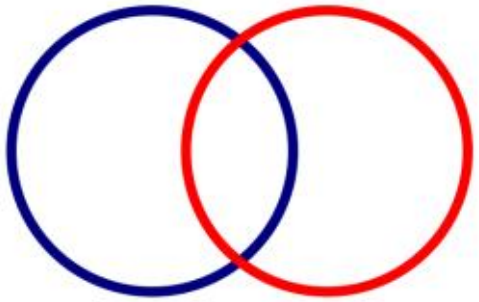
Mark uses Inkscape to create comics for the web (www.peppertop.com/) as well as for print. You can follow him on Twitter for more comic and Inkscape content: **@PeppertopComics**

Inkscape - Part 137

If the Page tool was the headline new feature for Inkscape 1.2, the equivalent for the recently released 1.3 version has to be the Shape Builder tool. It's modelled on the same feature in Adobe Illustrator, but I don't have enough experience with that program to be able to describe how the two tools might differ. Instead, I'll describe its usage from the assumption that you have some experience with Inkscape, but without further reference to its proprietary counterpart.

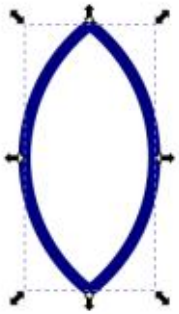
The Shape Builder can be thought of as a convenient way to perform certain Boolean operations between objects. There's nothing that can be done with the Shape Builder that wasn't previously possible with the existing tools, but anyone who has worked on complex designs using Boolean operations will be familiar with the need to plan multiple steps ahead, often duplicating objects that are otherwise removed as part of the operation. In many cases, the Shape Builder will greatly simplify such workflows.

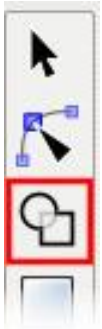
Let's look at the simplest of examples, in the form of a Boolean 'intersection' operation. For these first few examples, we'll just use a couple of overlapping circles, to which I've given thick coloured strokes to make them stand out. Be absolutely clear, however, that all these operations are taking place on the areas bounded by those strokes, not on the strokes themselves. This same behaviour would occur with thinner strokes, or even with filled shapes that have no stroke at all.



If we're starting with an intersection operation, let's first have a quick reminder of how to do this the Boolean way.

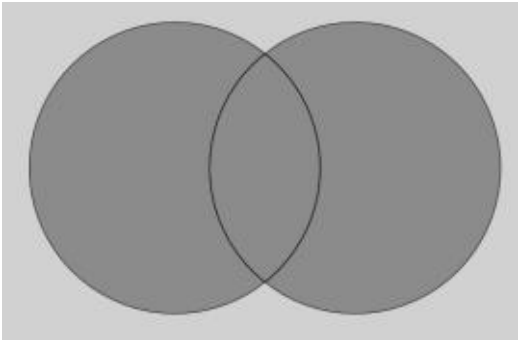
- Select both circles.
- Select the Path > Intersection menu entry.
- What remains is the common area between the two shapes – the part in the middle:





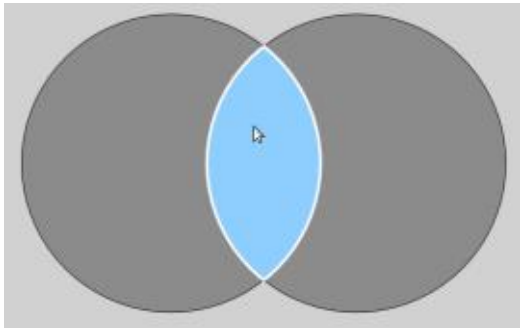
Notice that the resultant shape has adopted the style of the object on the bottom of the z-stack, resulting in a blue border. Now let's do the same thing using the Shape Builder tool, which can be found near the top of the toolbar, just below the Node tool. The default keyboard shortcut for it is 'X'.

As before, the first step is to select both objects that will contribute to the shape you're building. If you have nothing selected, Inkscape will present a small warning message when you switch to the Shape Builder. It is possible to use the tool with only a single object – though doing so only really makes sense if that object is a self-intersecting path. More usually, you'll want to select two or more objects before switching to the tool. When you do so, the original objects will be hidden and replaced with simple gray outline versions of the shapes. The rest of your drawing is also hidden, to avoid any confusion as to which objects are being used.

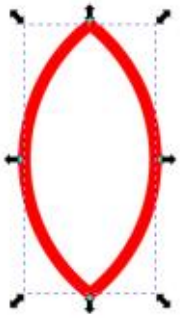


Moving the mouse over the shapes will highlight each individual section, rather than whole objects. In this case, therefore, there are three areas that can be highlighted as the

mouse moves over them, including the central piece we're interested in.



If you click on a section, it will change to a different shade of blue, indicating that it will be included in the final shape that you're building. In this case, we want only a single part, so click the central section, then press the Enter key to confirm that you're done. There's also a confirmation button in the tool control bar, and switching to the Selector tool also has the same effect. In either case, we're left with just the intersecting part of the original shapes.



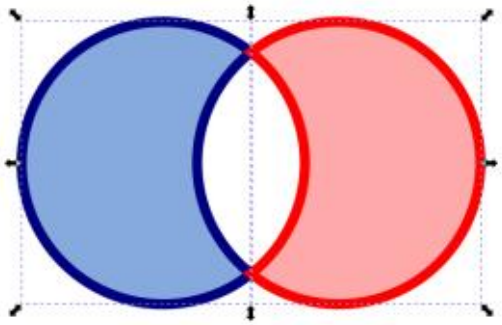
One important thing to note is that the outline of the shape is red this time, rather than blue. The Shape Builder uses the style of the topmost element in the z-stack to set the styles of the resultant object, not the bottom element as was the case with the Boolean operator.

Let's try another Boolean operation: Path > Exclusion, which will leave us with a shape consisting of the left and right parts of the two circles, but not the section where they overlap – essentially the exact opposite of Intersection. This time I've added some fill colors to the circles, to emphasise what actually remains.

Our final result is a path that consists of two sub-paths. That fact may not be entirely obvious, but it is possible to disentangle these two small moons from their kiss (e.g. with Path > Break Apart). The Boolean operation has done a very good job of cleanly removing the central part without introducing many new nodes.

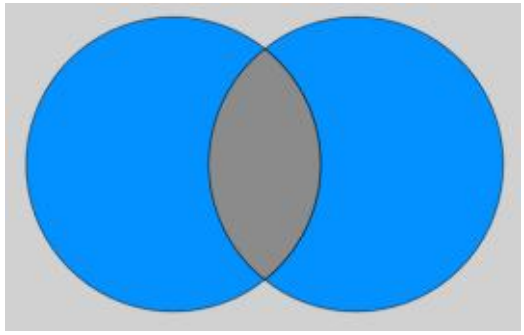
Doing the same thing with the Shape Builder is simultaneously both extremely simple, and rather complex. You see, for such a basic operation, there are actually several ways to proceed, depending on exactly what you want to be left with. The most obvious is to hold Shift while clicking on the central section in order to remove it from the final result, rather than adding it:

- Select both circles.
- Switch to the Shape Builder tool.
- Shift-click on the central section.
- Press Enter to accept the shape:



Well, it's kind-of the same thing. Except we don't actually have a single path this time, but rather our two original paths have both had a section cut from them.

Instead of Shift-clicking to remove the central section, an alternative approach is to click on both the left and right sections, leaving the middle one un-selected so that the Shape Builder view looks like this:

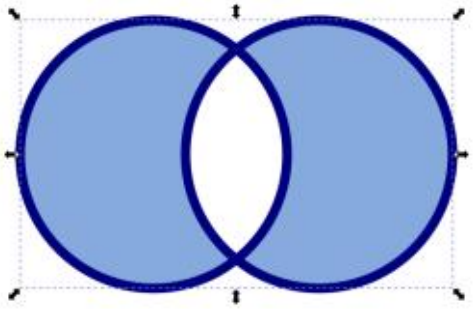


But, when you hit Enter, you'll find that you end up with exactly the same two objects as above, each with a section cut out.

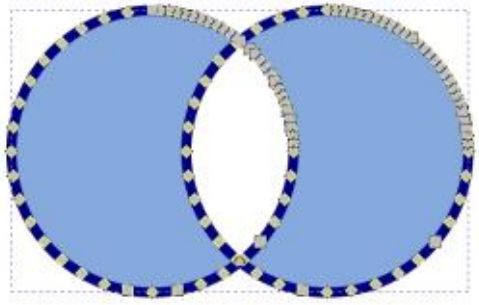
Suppose you really, really want to end up with a single complex path (i.e. one with sub-paths), but have an aversion to using Path > Combine for some reason. The Path Builder can actually produce a single path, but it requires a slight change of operation. Instead of clicking on each segment, you have to drag from one to another.

- Select both circles.
- Switch to the Shape Builder tool.
- Click and hold in the left-hand section.
- Drag the mouse to the right-hand section (avoiding the middle section).
- Release the mouse button.
- The two sections will be selected. Press Enter to confirm.

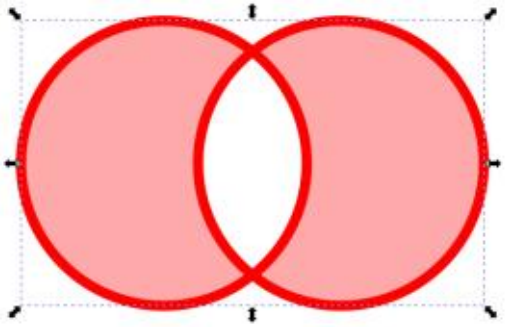
The result this time is that the selected sections have been combined into a single path, looking more like the Boolean version.



Looks can be deceiving, however. Switch to the Node tool, and you'll see that the Shape Builder has a tendency to create a lot of additional nodes. And I do mean a lot. This problem is already on the developers' radar, so hopefully version 1.3.1 might improve matters – but I still wouldn't be surprised to see more nodes from this tool than from the Boolean operations.



There's another interesting point to note when combining segments like this. Look at the result if I drag from the right segment to the left one, rather than left-to-right.

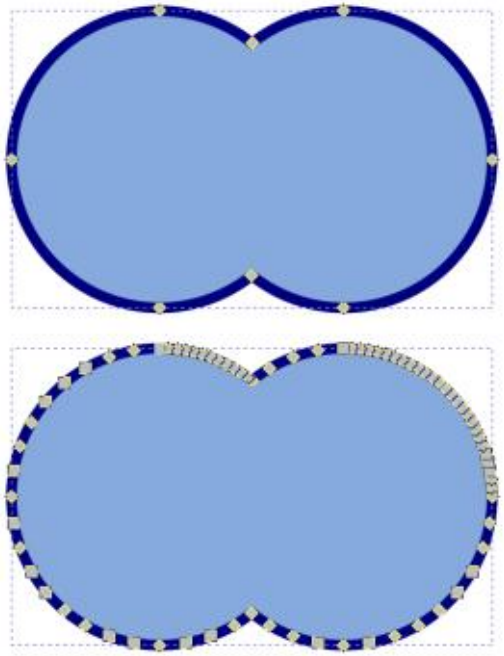


Looks rather more red, doesn't it? This provides us with another rule of Shape Building: when combining multiple segments, the style of the final result will be taken from that of the first segment selected.

The difference in behaviour between clicking segments and dragging over them means we can also use the Shape Builder to emulate the behaviour of another stalwart of the Boolean operations: union.

- Select both circles.
- Switch to the Shape Builder tool.
  Click and hold in the left-hand section.
- Drag the mouse through the middle to the right-hand section, selecting all three.
- Release the mouse button.
- Press Enter to confirm.

The result is that all the selected segments are combined into a single path. But although the results may look visually similar, again the Shape Builder leaves us with a lot of unnecessary nodes. This image shows the result of using the Boolean union (top) compared with the Shape Builder (bottom):

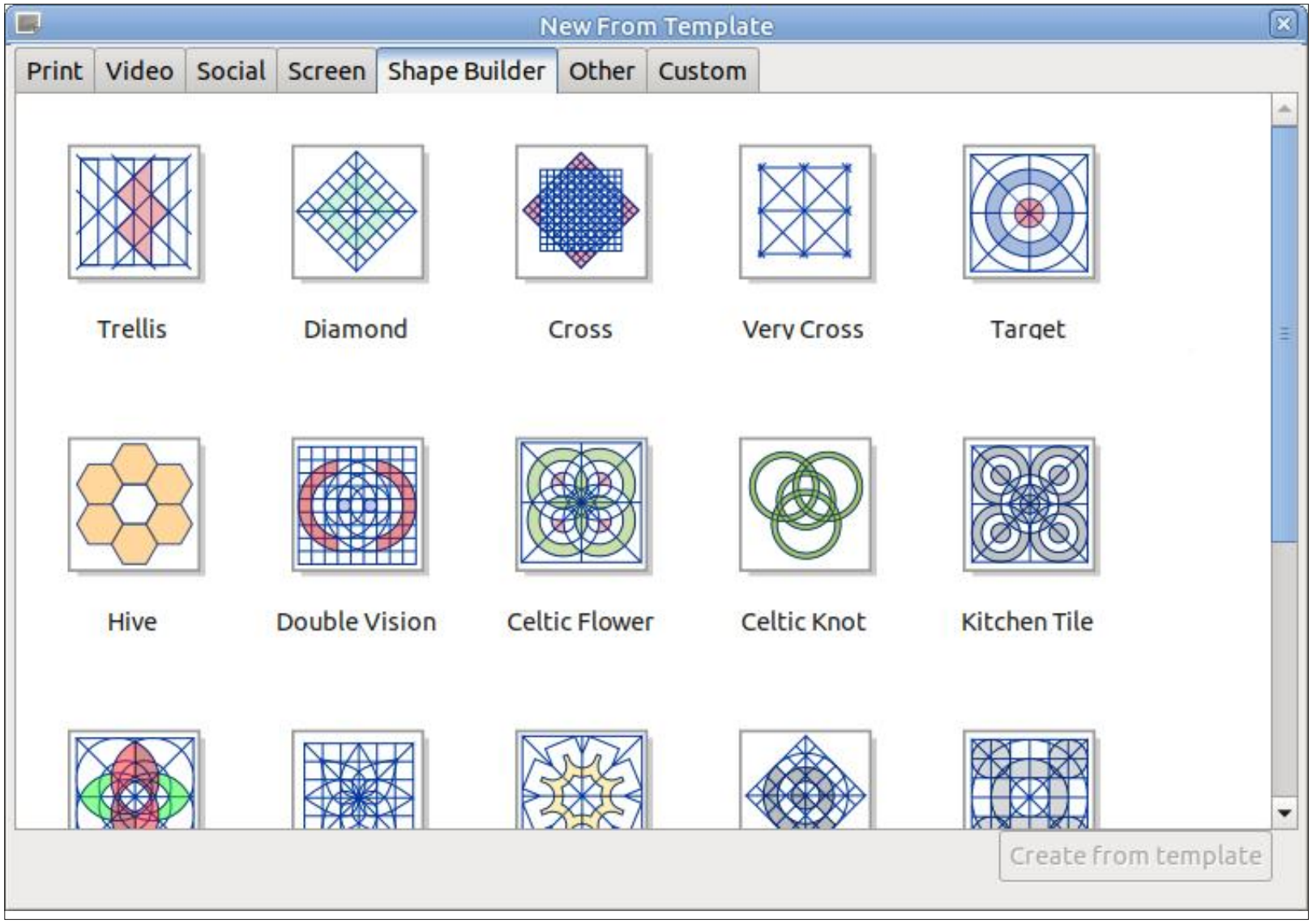


Given this obvious difference, you may be wondering what is the advantage of the Shape Builder when compared with the traditional Boolean operations. For simple operations that require only two objects, I think the Booleans, and related operations, are still the best way to go. But once you start combining multiple objects in complex ways, it's hard to argue against the simple click and drag approach of the Shape Builder.

One thing to be aware of as you work with more complex shape combinations is that you can use Ctrl-Z while in the Shape Builder tool to step back through your selections. This can be useful if you accidentally select or remove a segment you didn't mean to, and spot it immediately. If you don't see the error until you've made other selections, however, you're out of luck: there's no way to undo the selection or removal of an arbitrary individual segment. If you really aren't happy with the selections you've made, the Escape key, or the cancel button on the tool control bar, will abort the Shape Builder operation entirely. If you accept the shape, but you're not happy with the end result, a

quick Ctrl-Z will revert your objects back to how they were previously.

Finally, if you really want to get a feel for the power and flexibility of this new tool, why not use the File > New from Template… menu entry which opens a new page template dialog that includes a 'Shape Builder' tab filled with some complex overlapping paths that are ideal for experimentation.



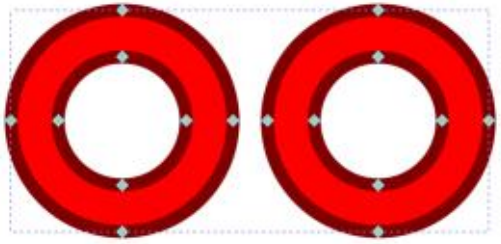
Mark uses Inkscape to create comics for the web (www.peppertop.com/) as well as for print. You can follow him on Twitter for more comic and Inkscape content: **@PeppertopComics**

Inkscape - Part 137

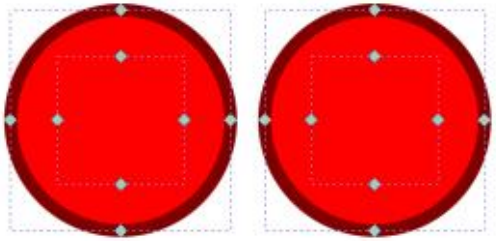
Last month, I took a look at the new Shape Builder tool in Inkscape 1.3, which offers a very convenient way to perform various Boolean operations. Unfortunately, it also tends to create a lot of extra nodes, so for some workflows, there's still a benefit to using the suite of Boolean operations under the Path menu. For users who wish to use those tools, I'm happy to report that Inkscape 1.2 and 1.3 both extended them a little.

Before I proceed too far, a note for the pedants: I'll refer to all these tools under the umbrella of "Boolean operations" for convenience. Technically speaking, the true Boolean operations are those that are based on Boolean algebra – a branch of mathematics named after George Boole, who formalised it in 1847. Only a few of the tools in the Path menu actually work in a strictly Boolean manner. But given that those tools are concerned with creating new shapes by combining and removing overlapping parts of multiple paths, it makes sense that other tools which achieve similar goals are grouped with them in the same menu. So for the purposes of simplicity, all the shape manipulation tools under the Path menu are Boolean operations as far as this column is concerned.

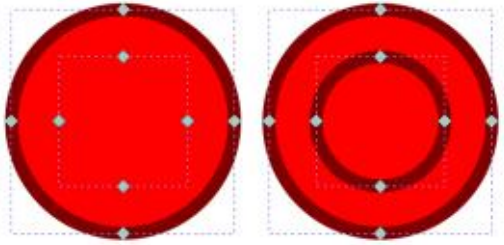
This month, we'll look at the first new addition, which was introduced in Inkscape 1.2: Path > Split Path. This is a little like the existing Path > Break Apart feature, except that it breaks paths apart in a way that probably makes more sense – and will be more useful – to most people. Let's start with a very simple example: here I've used Path > Difference to cut one circle out of the center of another, creating a donut shape. I've then duplicated the donut, and used Path > Combine (or Path > Union) to convert the whole arrangement into a single, complex path. As a quick reminder, I use the term 'complex path' to refer to any path which is made up of multiple sub-paths.



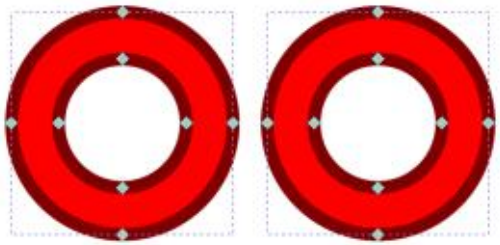
With the Node tool active, you can see from the image that there are four sub-paths, one for each dark-red circle, and each consisting of four nodes. Let's suppose that a little later we decide that it would be better to separate these objects in order to move them around independently. In previous versions, a naive user might reach for Path > Break Apart, only to be rather disappointed with the result.



Break Apart reduces a complex path to a collection of individual simple paths, by breaking each sub-path out into a separate object. In this case, we end up with four separate path objects, but because the larger paths no longer have a hole in the middle, their fill obscures the fact that there are a couple of smaller paths behind. Sending the outer circles down in the z-stack makes things a little more obvious – this screenshot leaves the left-hand circles untouched, but moves the large right-hand circle to the bottom of the stack.



That may have made it a bit clearer what has happened, but it doesn't really get us to the result we wanted – two separate donuts. For that, we have to select the two circles on the left and use Path > Combine, followed by doing the same to the two circles on the right. Finally we have two complex paths, each representing a single donut, which can be moved independently.



In this particular case, a user who is already familiar with the behaviour of Break Apart might have chosen to duplicate the original and cut half the content out from each copy using direct node editing, or some other Boolean technique that involves creating a quick sacrificial cutting shape. But for a more complex shape with lots of sub-paths, those approaches can quickly become unwieldy. Whichever approach you take, there's no doubt that it's a lot of steps just to split two visually separate objects into individual complex paths.

Now let's try the same thing using Path > Split Path. Starting with our complex path containing four sub-paths, applying this function gets us directly to the desired outcome (the same as the previous screenshot), with no additional fuss. Perfect!

Of course Inkscape isn't able to read your mind to determine how you want it to split your four sub-paths, and there are still cases where Break Apart might be the better option, so it's important to understand the logic used within this new feature. Quite simply, it breaks your path based on whether the sub-paths overlap or not. Any group of overlapping sub-paths will be split out into a new complex path. An individual sub-path that doesn't overlap with any other sub-path will be split out into its own new simple path. The net result is that a previously complex combination of shapes will be broken up into independent objects, based (broadly) on how visually separate they are.

An example with two donuts makes for a good introduction, but when would you actually want to use this feature in a real design? Perhaps the most common situation is when trying to split text (which has been converted into a path) into individual glyphs. In typographical terms, some characters have 'counters' – the closed shape in the middle of an 'o' or 'p'. These don't fare so well when Break Apart is used.

Use Path > Split Path, however, and the result is a lot closer to what you probably wanted.

Be aware, however, that it's not perfect. Inkscape doesn't know that this path represents characters, and Split Path just separates out the non-overlapping paths, so any glyph that is made up of non-overlapping parts will result in more than one path being created. You can see this with the 'i' or the '!' which are each split into two separate paths and need to be re-combined manually (with Path > Combine or Path > Union). It's still a lot less post-processing than with Break Apart, but don't think that the new feature is a complete panacea.

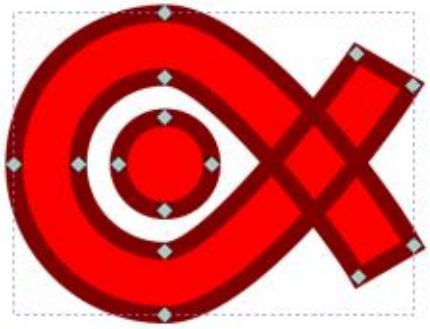
Another gotcha with this feature is Inkscape's notion of what constitutes a non-overlapping path. Consider this variation on our earlier donut, this time with an additional 'island' in the middle.

What do you think Path > Split Path will do in this case? It's pretty clear that the nodes for the island are all inside the outer sub-path, and therefore can be considered to be overlapping. Sure enough, Split Path has no effect on this shape at all. But what about this one?

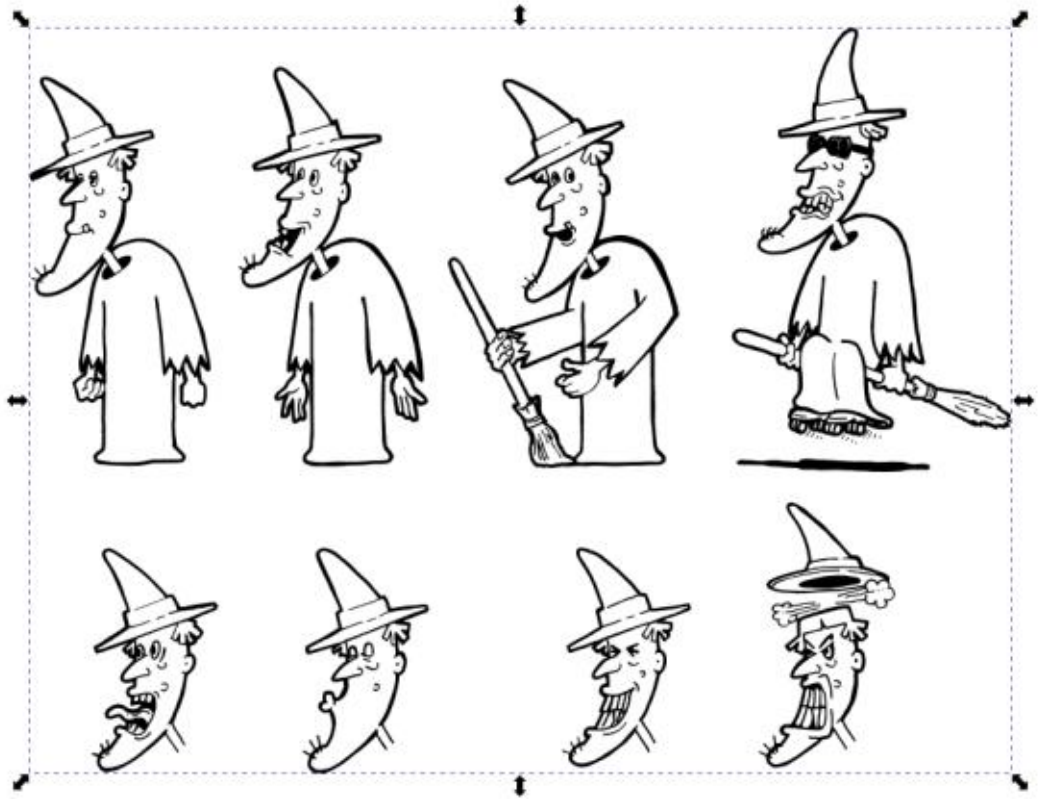
This time the outer shape has a tiny gap in it on the right-hand side. The thick stroke disguises it somewhat, but it's there. To the human eye, you might still consider the island to be 'inside' the shape, but, mathematically speaking, it's not. Inkscape goes with the

mathematical definition, so using Split Path results in two objects, not one.

Now let's extend that example a little further by moving the ends of the broken shape so that they cross.



From a human perspective the island is most definitely inside the shape this time. But, again, in mathematical terms, the two paths are not overlapping, so Split Path results in two separate objects. All of which is to reinforce the fact that this feature is more likely to give the desired result than Break Apart, but it still needs to be treated with care to ensure it has split things the way you want, especially when there are features that could end up as separate objects even if they give the appearance of being inside a larger shape.



I really wish this tool had been available a few years ago, when I was writing a weekly comic strip for my local newspaper. I created it in collaboration with Vince, a far more artistically competent friend. In order to turn the strips round quickly, he would draw the various elements of each comic separately on a sheet of paper, and I would then scan the entire sheet and trace it with Inkscape. Here's an example of a typical character sheet, after scanning and tracing.

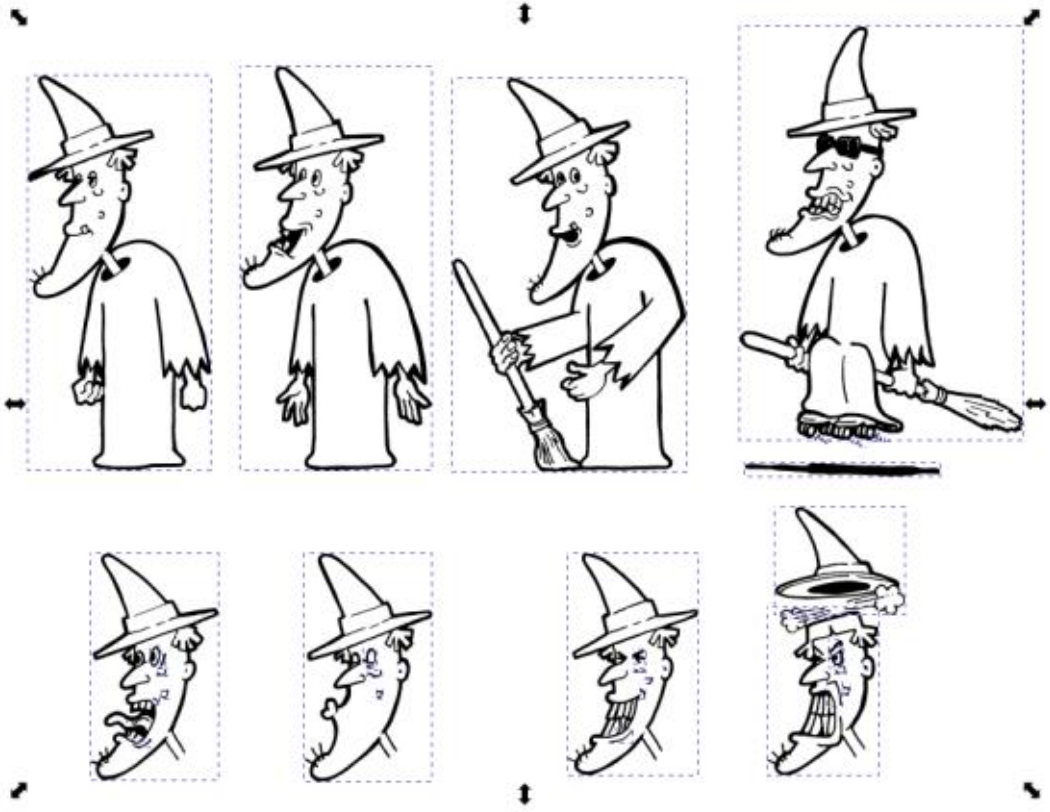
The problem here is that Inkscape's Trace Bitmap feature produces a single complex path for the entire traced image. In this case the path consists of over 15,000 nodes! Part of my job was therefore to create numerous duplicates of the traced image, each overlaid with a different cutting path. Path > Intersection was then used on each pair, removing all the parts that were not covered by the cutting path in each case. The final result was a separate complex path for each character or part on the sheet.

So how does Path > Split Path fare when faced with a task like this? Not at all badly, it turns out.

The first three images are perfectly cut out, and ready to go. The last one on the top row requires a small amount of re-combining to ensure that the shadow and the movement marks all remain with the character. The bottom row did a little worse, though it's hard to see from this screenshot. For each head there are some small elements – a wart here, an eye there – which have been split into separate paths. These are all due to the fact that the neck is not closed off, resulting in what Inkscape sees as an open shape, much like the broken circle I presented earlier. A little re-combining fixes the issue, but in reality we would have quickly learned that, while drawing, Vince should close the end of the shape in such situations, reducing the problem going forwards. Across dozens of character sheets and pages of props and backgrounds, Split Path would have saved me many, many hours of manual path tweaking.

If you frequently work with complex paths, text-as-paths, or bitmap traces, I hope you can see how this feature might be a game-changer for your workflow. Compared with the alternatives, it will often do a better job with less complexity and confusion. Even in those cases where the results aren't perfect, they'll probably be more understandable to most users than the confusing mess that Break Apart often appears to make of things.

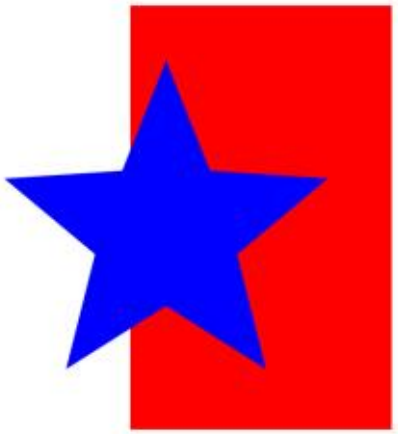
Next month, I'll continue this theme, to take a look at the new Boolean operations that were added with Inkscape 1.3.



Mark uses Inkscape to create comics for the web (www.peppertop.com/) as well as for print. You can follow him on Twitter for more comic and Inkscape content: **@PeppertopComics**

Inkscape - Part 139

Last month, I spent the entire article looking at the Path > Split Path operation that was added in version 1.2. Although version 1.3 added two more path operations, they're essentially two variations on the same theme, so won't take quite as much space to describe. These are Path > Flatten and Path > Fracture. Let's begin with an example that consists of a star drawn over a rectangle:



You probably give little thought to such layering of elements, which you undoubtedly use all the time in your Inkscape projects. But to understand how these new operations work, it's important to comprehend what's actually happening in terms of the SVG content. Both these objects are in the same layer, but they live at different positions in the Z-stack, usually determined by the order in which they appear in the file. This is the "painter's model" that SVG uses – earlier objects in the file can be "painted over" by later objects. In this case, the blue star is painted over the red rectangle, and since there's no transparency involved, we see a solid blue color even in the overlapping regions. If we were to imagine looking at this arrangement from the side, it might look something like this:

I've drawn this to look as though the star is casting a shadow on the rectangle. In practice that purple shadow actually represents the area of overlap between these objects.

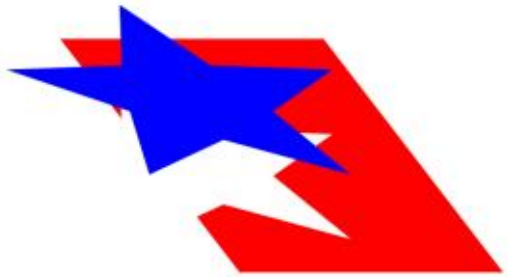
So far, I probably haven't told you anything you didn't already know, even if that knowledge doesn't usually play an active role in your use of Inkscape. Indeed, for most people, the way that the shapes are 'layered' is purely an academic consideration: in practice, Inkscape draws solid blue pixels on the screen and you don't need to concern yourself with the fact that they're actually obscuring some red pixels from the rectangle behind. But there are a few situations where this knowledge is vital.

Consider screen printing – often used for printing designs onto T-shirts, posters and fabrics in general. It's somewhat fallen out of fashion now, as on-demand printers can put your full color design onto all manner of items without the hassle of setting up your own mini print shop. But for artistic, budgetary, or other reasons, screen printing still lives on. In this process, each color in the design is printed by forcing ink through a mesh screen that carries the part of the design showing that one color alone. Repeat this with a different screen for each color, and you can reproduce complex designs, provided they only have a limited number of colors.

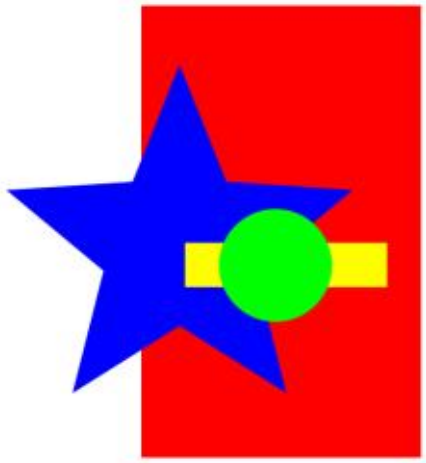
In the case of our design above, a naive approach would be to create one screen with a rectangle, and one with a star. First use the rectangle mesh to print in red, then align the star mesh and print the blue parts. But in this case we're not dealing with pixels in memory that don't exist until the final rendering step – we're dealing with wet inks that will merge and run into each other. Our final design won't show a blue star and red rectangle, but rather parts of each, with a muddled purple area where the shapes intersect.

What we actually want is for that overlapping area to be removed before printing. We want the design (when viewed from the side) to look more like this, with the intersection cut out of the red rectangle:

In this particular case, with just

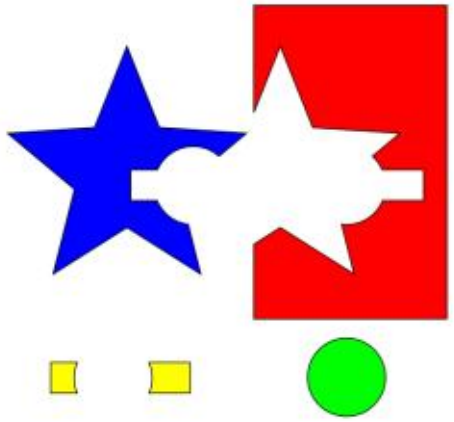


two objects to consider, it's not too tricky to duplicate the star and use Path > Difference to cut it out of the rectangle. But how about when three or four objects overlap – let alone more.



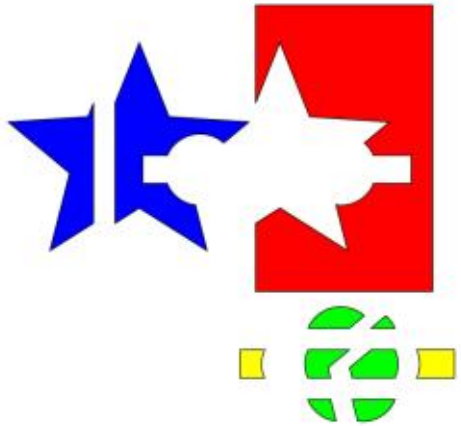
Just adding a yellow rectangle and a green circle into the mix starts to make the use of Path > Difference rather complex. The green circle has to be cut out of all three objects below it, the yellow rectangle out of the two below it, and so on. The more objects, colors or complexity, the harder it is to manually perform all the operations required to produce a final result with no overlapping parts.

What is needed is a simple way to modify the image so that the final result consists only of the visible parts of paths, with no overlapping sections. In raster graphics programs, this is a common task for combining multiple layers into one, where it's referred to as 'flattening' the layers. And so, with Inkscape 1.3, we now have Path > Flatten to achieve the same effect with paths. Selecting all four paths in this example and applying this operation results in the following four objects (moved apart, and with strokes added for clarity):



this will be fine, and represents a much faster way of achieving what would previously have been a tedious and error-prone series of Boolean operations.

The other new path operation does something similar, but breaks elements apart even further. When using Path > Fracture, you not only get the flattening effect, but the overlapping shapes are further broken apart as though some Path > Division shenanigans had also taken place. You can see how, in this example, it results in far more individual paths than the Flatten operation (again, moved apart and strokes added for clarity):



To be honest, I haven't yet thought of a good example of where flattening and splitting paths in this way would be useful. But perhaps that says more about my lack of imagination, and this feature might be just the thing you've been waiting for to revolutionise your Inkscape workflow.

While we're dealing with the Boolean operations there's another change in 1.3 that needs to be discussed: what happens when you use Path > Object to Path with a text object. If you're getting a sense of déjà-vu, it's because this is a topic that has cropped up previously in this series, as the Inkscape developers seem insistent on modifying the behaviour every few releases.

For your average screen printer,

Up to version 0.47, this operation simply converted the entire text content into a single complex path. This made it extremely difficult to work with the individual characters (technically, glyphs), if that was your goal. Version 0.48 changed the behaviour to create a single group consisting of one path per glyph. This made some tasks a lot easier, and if you really did want just a single path, using Path > Union rather than Object to Path would still achieve that without having to ungroup and combine separate paths. All was well, until version 1.0 broke the Path > Union trick… but the developers fixed it once more in 1.0.2 (see part 100 of this series for more details).

So, aside from a brief period after version 1.0 was released, this functionality has been pretty stable: Path > Object to Path creates a group containing one path per glyph, while Path > Union creates a single path for the entire text object. Everyone was happy, and there was definitely, absolutely, no need to upset that status quo, right?

Apparently the Inkscape developers either didn't get the memo, or there's a secret cabal of disruptive users who never really got over the change from v0.47, because version 1.3 brings back the bad-old days when Object to Path created a single path for the entire text, with no option to create separate paths for each character. But wait! Don't forget last month's column, where I looked at the Path > Split Path operation. Surely that can help. Well… maybe. Sometimes. Sort of.

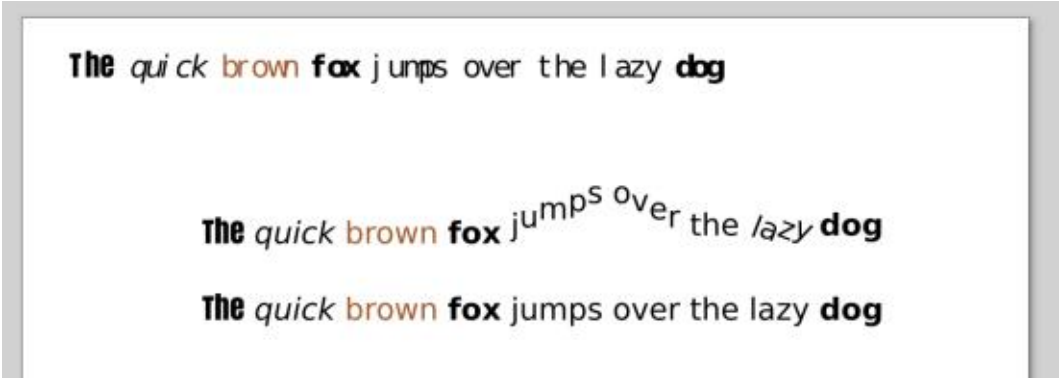
To use Path > Split Path you first need a path to split. Unfortunately it won't auto-convert a text object on your behalf, so you have to use Path > Object to Path first, and then follow it up with Path > Split Path. However, as I noted last month, as useful as the Split Path function is, it doesn't understand that you're working with glyphs. The dot over every 'i' and 'j' becomes a separate path object, as do accents over characters, or the dots at the bottom of question and exclamation marks. If you're lucky you might get away with using this function directly, but more often than not, there will be additional manual work required to re-combine the disconnected parts of such characters.

There's a "solution" to this problem which should be present in the 1.3.1 release (which will probably already be out by the time you read this). This version adds a Text > Text to Glyphs menu entry, which can be used to split a text object into individual glyphs before you use Path > Object to Path on them. I've tried it in the 1.3.1 Release Candidate build, and it works… but it's still adding an extra step that wasn't necessary before. If you're still on version 1.3, you may be able to use the Text > Split Text extension (in the Extensions menu) to achieve the same result – though my own experience with this has been extremely poor, with the split characters being badly misplaced.

Speaking of badly misplaced characters, the new Text to Glyphs function also moves your text around if you've adjusted the vertical position or the rotation of individual characters.

The example below shows the results of both the extension, and the new function. The text in the middle is the original: I've deliberately used two fonts, with one of them in different weights and styles. I've also manually adjusted the vertical height of some of the letters, and the rotation of others.

The line at the top is the result of using the extension. To be clear, I haven't moved it to that location – the extension decided to place the result at the top of the page, ignoring the position of the original text object. It's done a good job of preserving the fonts, weights and styles. But not only has it ignored the vertical adjustments and the rotation, but it also has very odd ideas about the spacing between



characters.

The bottom line shows the result of the Text to Glyph function. This time the split text appeared in the same location as the original, so I have moved it down. You can see that the fonts, weights and styles have been preserved, but vertical alignment and rotation have, again, been ignored. Of the two, however, it definitely gives the better result.

Let's compare this to the behaviour of 1.2.x. In this case, the original text is at the top, and two copies have been made and moved down so you can see the result of each operation more clearly. The second line is the result of Path > Object to Path. As you can see, it looks identical to the original in both style and position. But in practice, this is now a group of individual paths, one for each glyph. The third line shows the result of Path > Union, which again preserves the style and position, but loses the color change due to having created a single complex path for the whole text.

In my opinion this change in behaviour is a massive step backwards. It totally removes some perfectly reliable functionality from 1.2, replacing it with options that are far less functional – but it doesn't appear to offer any new capability that makes this trade-off worthwhile. If you ever play around with the alignment and rotation of individual characters in your text, the only way to create a group of paths from the carefully-placed glyphs is now to use Object to Path, then Split Path, then manually fix up any characters that are made up of multiple parts. But you'll also lose any color changes along the way, and will have to re-apply those manually as well. It turns a single step operation into something vastly more complex.

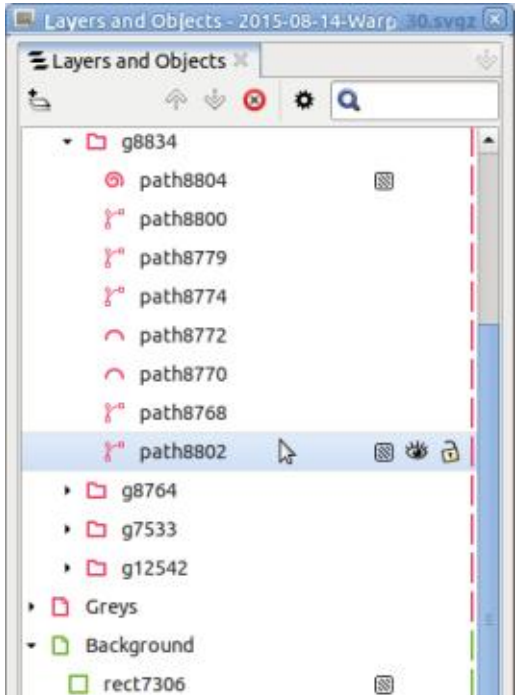
I wonder how long it will be before I'm writing a column to describe yet another change in this behaviour…?

Mark uses Inkscape to create comics for the web (www.peppertop.com/) as well as for print. You can follow him on Twitter for more comic and Inkscape content: **@PeppertopComics**

Inkscape - Part 140

For a long time users were crying out for Inkscape to have a single dialog that lists all the objects in a document. So the developers added one, back in version 0.92, but it did have an annoying side-effect of slowing the application down once the dialog was opened – even if you then closed it again! Quitting and restarting the application solved it (provided you didn't open the dialog again), but it was frustrating enough that this feature wasn't terribly useful on more complex projects, even though those were the ones that would most benefit from an Objects dialog. Meanwhile the long-standing Layers dialog was still performing its task admirably, but was obviously limited to showing a list of layers.

With version 1.2 both these older dialogs were removed, and their functionality combined into a single 'Layers and Objects' dialog. In order to avoid too much confusion for long-time users, it's available from both the Layer menu and the Object menu, so if you were used to opening either of the older dialogs from the menus you'll still manage to find the new one. Similarly the old Layers button in the main toolbar will open the dialog, as will the Ctrl-Shift-L keyboard shortcut. One small annoyance, in my view, is that the tooltip on the toolbar button describes it simply as 'Open Objects' and doesn't mention layers at all. This is particularly jarring as the 'L' in the keyboard shortcut is a lot easier to remember if you think of it as the 'Layers and Objects' dialog.



The image below shows the dialog as it appears in version 1.3. The key features are present in version 1.2, but the later release adds a few extra bits that are worth discussing.

Every object, group or layer in the document gets its own line in this dialog, which shows the following items:

- An indicator to show whether the layer or group has been expanded
- A colored symbolic icon representing the type of object
- The label of the object
- An icon indicating the opacity and blend mode of the object
- An icon to indicate whether or not the object is visible
- An icon to indicate whether or not the object is locked
- A thin swatch whose color matches that of the symbolic icon

Item 1 only appears on layer or groups, and only those that contain other objects. Items 4, 5 and 6 are visible as you hover over a row with the mouse. Clicking on these three in order to change the defaults can result in them remaining visible even when the mouse has moved away – I'll describe that in a bit more detail below.

The colors assigned to each topmost layer are drawn from a palette of 8 values, based on the label of the object. Try creating layers named "1" to "8" to see what colors are used in your chosen theme – and throw in a "9" or a "0" if you want to prove there are only 8 in use. Here are the default colors on my system:

The fact that these colors are assigned based on the label means that you may end up with identical colors next to each other. Changing one of the labels a little may fix that, but if it really bothers you then you can manually set the color by clicking on the thin swatch at the right of the row. More on that shortly.

Eagle-eyed readers will have

noticed that I said "the colors assigned to each topmost layer", rather than to each object. The labels of individual objects and groups – and even of sub-layers – have no bearing on the initial color that is applied. That comes entirely from the color of the topmost layer for that part of the tree. This leads to all the objects within each top layer being given the same icon color, which can be a bit of a problem if you're the sort of person who keeps the whole structure of your drawing inside a single top-level layer.

To help address this, it's possible to override the default color for each item in the tree by clicking on the thin swatch at the right. This brings up a small color-picker dialog in which you can change the icon's color. If you do this on a layer, sub-layer or group, any elements created within it will be given the new color for their icon. Any existing descendent objects will also be updated to the new color, unless they've been explicitly overridden (or they're in a sub-layer or group that has). The dialog provides no way to remove an override and revert back to automatic assignment. If you really want to do that, it will require a trip to the XML Editor to remove the 'inkscape:highlight-color' attribute.

Name	Value
inkscape:highlight-color	#ff422f
inkscape:label	en-GB
id	layer8
inkscape:groupmode	layer

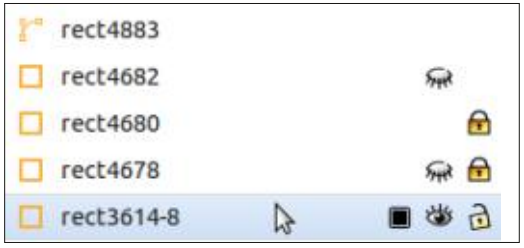
One rather annoying aspect of the color-picker dialog is that it affects the row whose swatch was last clicked, not the currently selected row. So if you wish to change the icon color for several items you will need to do it one-at-a-time, clicking on the swatch for each one. You don't have to close the dialog in-between each of them, but it's still rather annoying if you want to set multiple items to have the same custom color. In practice I doubt many people bother setting custom colors anyway, so it's not a huge problem – but it would be nice to be able to use some of the advanced features of the Find/Replace dialog to select multiple items, then set them all to something that stands out prominently in the list.

Let's move past the color of the symbolic icon and onto its shape. The choice of icon reflects the type of Inkscape object you're looking at. The specific icons vary between themes but, on my system, layers and sub-layers appear with a page icon, groups with a folder icon, text with a "T", and so on. If an object is clipped or masked, it also gains a small badge on the icon to indicate that state – and if a clipped object has a masked applied, or a masked object is then clipped, you do get both badges.

It's very important to note that the icon used reflects the current type of the object, which may not be how it was originally created. A rectangle that has been converted to a path, for example, will have a path icon, even if the label is still the default "rect1234" type of string.

Speaking of those labels, this dialog is perhaps a more useful place to set them than via the Object Properties dialog. Just double-click on a label to edit it in-place. Don't forget that changing the label of a top-level layer may also change the color of its icon, and those of its descendants. Do note, however, that the label is an Inkscape-specific thing that is purely there for your convenience. I tend to set the labels on layers and perhaps a few key groups or objects, but it's rarely worth trying to set sensible labels on everything. It also doesn't change the XML ID of the object, so JavaScript developers who want to use specific IDs for their code to reference will still need to open the Object Properties dialog or the XML Editor.

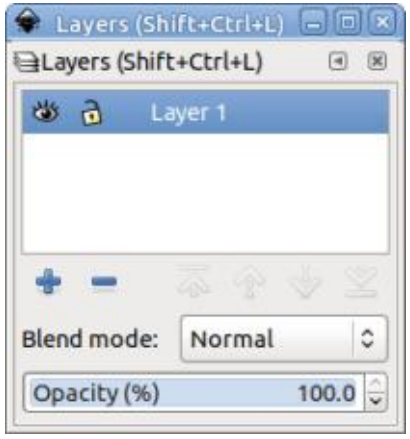
Moving on to the three icons that appear when you hover over a row, I'm going to skip the first one and come back to it shortly. The second and third are used to indicate the visibility and locked state of the object. The default for any object is for it to be both visible and unlocked, so Inkscape doesn't bother to show the icons on rows where that's the case. The only exception is the currently hovered row, where those icons also act as buttons to toggle the state. If you make the object invisible or locked, the corresponding icon remains visible even when you're no longer hovering over the row, making it



easy to see which objects are in the non-default state.

If these icons are toggled on a layer or group, then the descendants gain a translucent version of the default icon, indicating that even though the defaults are used, their final appearance is being determined by the ancestor further up the tree:

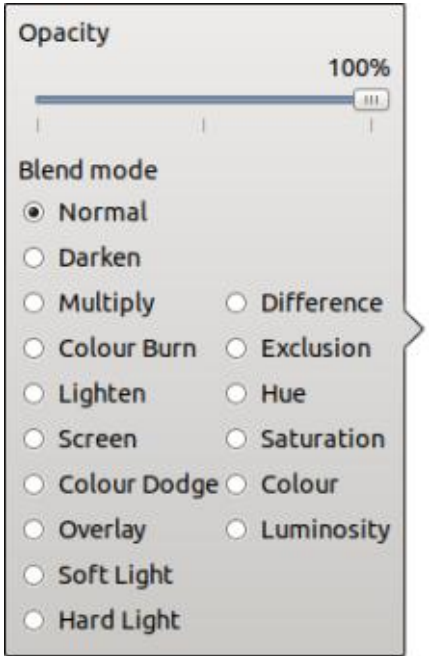
Much like these two controls, the first icon of the three is also hidden when the object is in its default state, but this is not a simple on/off toggle like the others. This control is used to indicate and change both the opacity of the object, and its blend mode. Before describing this in more detail, it's worth considering why such a control has been added, by taking a look at the Layers dialog from version 0.92:



At the bottom are convenience controls for selecting the layer's blend mode and opacity. To be honest, I would imagine that the blend mode popup was rarely used beyond a little experimentation. Blend modes can definitely have their uses, but although they can be used to great effect in a bitmap editor, I've hardly ever found a need for them in a vector graphics program. Nevertheless, having them there definitely made them far more discoverable to users than their other home inside the Filter Editor.

With the new dialog in 1.2 these controls didn't make the cut. The advice to users who wanted to use blend modes was to add them via the Filter Editor, while opacity could be set in the Fill & Stroke dialog, even for layers.

In 1.3 those controls have made it into the new dialog, in the form of that first control in the trio. Personally I would have made it a quartet of controls, and kept the Opacity and Blend Mode as separate things but, instead, clicking on that icon presents a pop-up which combines both of them, looking like this:



As you can see, this pop-up allows you to adjust the Opacity using a slider, and choose the Blend Mode using a collection of radio buttons. Frustratingly there's no numeric input for the Opacity slider, so if you want a specific value you may have to manage it via the Fill & Stroke dialog anyway. Incidentally, if you need more information about blend modes, I covered them well over a decade ago, in part 9 of this series (Full Circle Magazine #69) and aside from the change in UI the information there still holds true.

The default values for an object are 100% opacity and Normal blend mode. The icon will not appear for any row with those settings unless it's being hovered over. Reduce the opacity or change the blend mode, however, and an icon will be shown, even for non-hovered rows. Considering the states of both the opacity and blend mode, there are six possible combinations that can be shown. Here's how they appear with the theme I use:

The first, a fully filled single square, is what you'll see when hovering over an object with the default settings. If you reduce the opacity to anything less than 100%, but greater than 0%, the second icon will be displayed. The third

appears for objects with 0% opacity. I really do mean 0% - even a 0.1% opacity, despite appearing completely transparent to the human eye, will be displayed with the second icon.

If you change the blend mode, but leave the opacity at 100%, you'll get the fourth icon. The fifth is for a non-Normal blend mode and less than 100% (but more than 0%) opacity. There isn't a specific icon for a non-normal blend mode with 0% opacity, and the icon from row 3 is re-used. I guess this makes some sense, because a blend mode doesn't really have an effect on a fully transparent object, but I'd still prefer to see that state called out explicitly.

Each row in this dialog also has a right-click context menu that is identical to the one you would get when right-clicking on the same object on the canvas. One handy shortcut to remember is pressing '3' to zoom to fit the current selection in the window: just click on an object or layer in the dialog and press 3 to bring it into view.

The last thing to mention in this dialog is the toolbar at the top. Version 1.2 had a simplified version of this, lacking the search box and settings button. But it did start with a toggle to switch between a view that includes all objects and one that shows only layers – the latter mode mimicking the Layers dialog from earlier releases for those users who never really found much need for a list of every object in the document (I'm one of them). This option is still available, but has been moved into the Settings pop-up, which unfortunately makes it a little less practical to switch back and forth between the two modes as the need arises.

The second button on 1.2, or the first on 1.3, is used to create a new layer, opening the same small dialog as we're used to from earlier releases. The Up/Down buttons shift the selected object(s) up and down the list – which also moves them up and down in the z-order. When multiple items are selected, each is moved relative to its siblings, but the entire operation is prevented if any one of them is unable to move any further. Then there's a delete button which will delete the selected object(s) or layer(s) immediately.

The Settings button has the 'Only show layers' option described above, but also an 'Expand to display selection' option. I recommend leaving this one enabled, as it causes the dialog to automatically expand any layers, sub-layers and groups that are required to ensure the selected object is visible in the tree. Turning this off stops the tree from 'jumping around' so much, but can make it less obvious which object is selected.

Finally, version 1.3 introduces a search box. Typing something into this and pressing Enter will filter the list to show only objects that match the string. The search is case-insensitive and will match a substring, and the test is performed against not only the label, but also the ID. No other attributes or text nodes are searched, so you can't use this to filter by color or the contents of a text object, for example. To return to the complete list of objects, use the button in the search field to clear it – there's no need to press Enter in this case.

The addition of this feature is undoubtedly an improvement over the old Objects dialog, and it does make a lot of sense to have a single UI that shows both objects and layers. There are a few small design choices that I might have made differently, but I definitely take my hat off to the Inkscape team for completely replacing the old dialogs in an effort to push the application forward.

Mark uses Inkscape to create comics for the web (www.peppertop.com/) as well as for print. You can follow him on Twitter for more comic and Inkscape content: **@PeppertopComics**

Inkscape - Part 141

First this time a little update on Inkscape versions. I mentioned a couple of months ago that a release of Inkscape 1.3.1 was expected, which adds the Text > Text to Glyphs feature. This was released on schedule, and also adds the welcome addition of a new entry in the Snap popup which allows you to turn off snapping to grid lines while still allowing snapping to grid intersections. On top of that there were a lot of bugfixes – way too many to document here, so take a look at the release notes if you're interested: https://wiki.inkscape.org/wiki/Release_notes/1.3.1

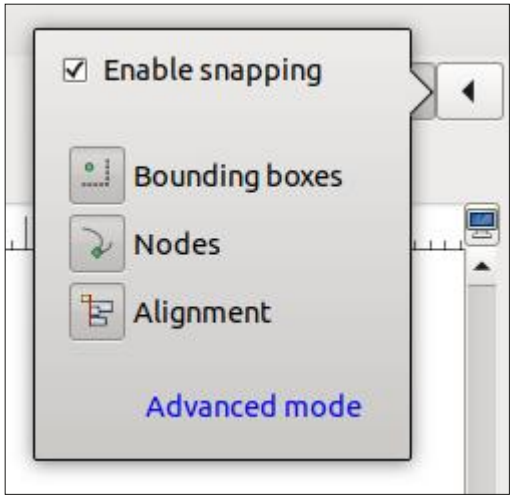
But before you rush off to download 1.3.1 you should be aware that it also introduced some bugs that can lead to data loss or corrupted files. As a result, version 1.3.2 was quickly released to address these issues, and this version is a strongly recommended update for anyone using 1.3. If you've already downloaded 1.3.1 then the new release should be considered an essential upgrade. As usual, the application can be downloaded directly from https://inkscape.org

I'm still working my way through the bigger updates and changes that came with versions 1.2.x and 1.3.x. According to my notes, most of the things I still have to cover were added with 1.3, but there are a few stragglers on the 1.2 list. To keep my own housekeeping simpler I'm going to rattle through the outstanding 1.2 features this month, clearing the boards to concentrate on 1.3 going forwards. This will result in a bit of a hodgepodge of topics that don't necessarily relate to one another in any manner other than the release version, and even though they were added in 1.2, I will be describing them as they currently appear in version 1.3.2.

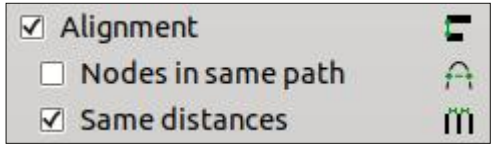
Alignment & Distribution Snapping

This is a feature that has been common in other vector graphics programs for some time – particularly those used for user interface mock-ups. This mode adds dynamic snap points that are derived from the positions of other elements in the page – a fancy way of saying that it makes it easier to ensure that items are neatly aligned or spaced as you drag them around.

There are two aspects to this feature: aligning to existing objects, and spacing evenly from nearby objects. By default these are both disabled, and need to be turned on from the Snap popup. If you use this in its simple mode, then you just need to switch on the 'Alignment' option at the bottom:



If you use the advanced mode of the Snap popup then you'll need to enable both 'Alignment' and 'Same distances' to enable both types of snapping. Note that the UI here is misleading, as it suggests that turning off 'Alignment' will also turn off 'Same distances' but that's not really the case. Doing so will certainly disable the control in the popup, but its current state will still apply, meaning that it is possible to have 'Same distances' enabled while 'Alignment' is disabled, should you need to, with a bit of checkbox juggling.



With both those options enabled, what effect does it have? Unfortunately this is a feature that is much more easily demonstrated with a video than a screenshot, but I'll do my best. Consider a simple arrangement of three squares to which I wish to add a fourth one:

With this feature enabled, as I drag the blue square towards the others, various extender lines will be projected as the box becomes aligned with the center or edges of the existing elements, while other lines will appear between the items when the position of the blue box matches the spacing between the red ones. Here's how it looks as I drag the blue box into place:

Note the line running from the center of the last red box to the center of the blue one, indicating that this snap position has the centers vertically aligned; and the three spacing lines between each of the boxes indicating that this snap position will put the blue box equidistant from its peers. Achieving such alignments was already possible, but nowhere near as intuitive as with this system. It's not quite as smooth and seamless as the same feature in some other applications, but it's a welcome addition even in its current state.

As I mentioned at the outset, this is a particularly common feature in UI mock-up tools, where it's used to quickly line-up buttons, labels and other controls in a way that makes the result look more professional. This addition undoubtedly improves Inkscape's capabilities in that regard. Consider this mock-up of a 'card' element for a web page, with a few arbitrary UI controls in it: previously just sorting out the alignment of the various elements would have taken as much time as producing the rest of the layout. Now, it's trivial:

When using the Node tool to manually adjust node positions, it's also possible to enable similar snapping to existing nodes within the same path. This can only be done via the advanced mode of the snap popup, by checking the 'Nodes in same path' option. Note that this adds extension lines and snapping to the horizontal and vertical positions of other nodes, but doesn't provide the same guides for equidistant spacing as you see when moving whole objects.

Node Tool improvements

While we're on the subject of the Node tool, this saw some small but useful additions with 1.2. You're surely already aware that this tool allows you to select multiple nodes within a path by drawing a 'rubber-band' box that encompasses them. Sometimes this results in too many nodes being selected, and in the past you had to then deselect any unwanted nodes one-by-one, by clicking on them while holding Shift. Now, however, you can also draw a rubber-band box to deselect nodes en masse, by holding Shift and Control while you drag out the rectangle.

It's also now possible to use the rubber-band mode to perform an 'inverted' node selection – the selected nodes will be all those that were outside the selection rectangle. Just hold Ctrl as you drag out the rectangle to achieve this. You could achieve the same previously by simply using Edit > Invert Selection after selecting the nodes, but just holding Ctrl while dragging is a slightly easier solution. There is one difference in behaviour, when dealing with complex paths: whereas Edit > Invert Selection could be used to invert the selection state of nodes in a single subpath, leaving other subpaths unaffected, the inverted rubber-band selection results in all the nodes outside the rectangle being selected, across all the subpaths.

Set origin of Selector tool transformations

Switching to the Selector tool (F1), this gains a new ability that makes it easier to precisely position and scale items using the numeric fields in the tool control bar. This has always been possible, of course – who hasn't adjusted the 'W' and 'H' fields to draw a rectangle of a specific size? But previously any such adjustment used the top-left of the bounding box as the reference point for any changes.

(Note: In pre-1.0 releases the reference point was actually the bottom-left, but that version introduced a flip in the default orientation of the y-axis. If you switch back to the older orientation via the preferences then the reference point also switches back to the bottom-left).

Whether it was top-left or bottom-left doesn't really matter: the fact is that a single reference point was used. Type in some X and Y coordinates and that's where the reference point would be placed. Adjust the width and height, and the reference corner would stay put, while the changes propagated out from there.

1.2 added the ability to set one of 8 different reference points for such transformations. With the Selection tool enabled, just click once on any of the 8 resize handles (the arrows that surround a selected object) to set that point as the reference. The arrow will go a very non-obvious shade of dark blue (at least on my theme), and barely perceptible lines will be projected across the canvas to indicate which corner or mid-edge is the current reference point. Click on the handle again to return to the default reference point.

This is another welcome addition, but with one caveat: it's a shame there's not a way to select the center-center point right in the middle of the bounding box. If you wish to position an element so that its center is at a specific set of coordinates, you will need to first select the mid-point handle for the top or bottom side, then set the x coordinate, then select the mid-point handle for the left or right side, then set the y-coordinates. In fact you might find it faster to draw a temporary rectangle, set the coordinates so that the top-left corner is in the desired position, then snap your real object to it using the alignment snapping mode described above.

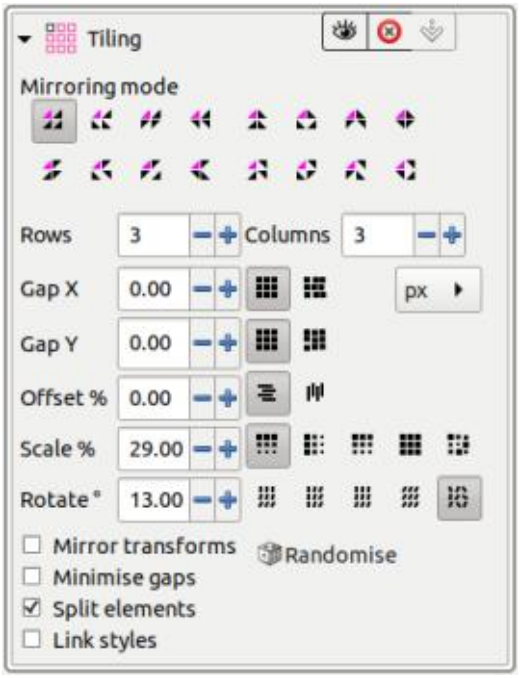
Tiling LPE

A new LPE called 'Tiling' has been added, which provides similar functionality to the existing Edit > Clone > Create Tiled Clones… dialog. The biggest difference is that the output in this case is a complex path, rather than a collection of clones. It can also be a number of separate paths, if the 'Split elements' checkbox is enabled… but they're still not clones in the traditional Inkscape sense (<use> elements in SVG terms).

The new LPE is certainly more interactive than the older dialog and arguably a little more intuitive (though not by much, to be honest). Note, however, that some of the more advanced aspects of the Tiled Clones dialog are not available in the LPE. If you just want to create a grid of elements, perhaps with a bit of variation or randomisation of positions and sizes, then this LPE might be just the thing for you. For my needs, I'll be sticking with the Tiles Clones for now, but definitely keeping an eye on this extension to see if it is developed further in future releases.

I don't intent to describe this LPE any further, given that I've previously described the Tiled Clones dialog in a lot of detail (FCM #93 to #96). Many of the general principals are the same, and a lot can be inferred from trial and error. But if you do want me to go over this one in depth, drop a line to the magazine to indicate your interest.

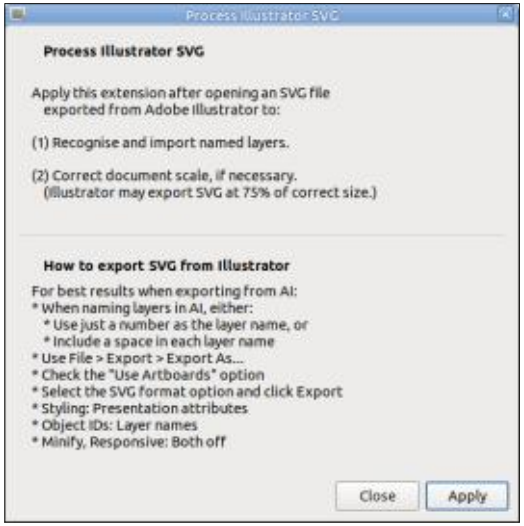
Extensions



Version 1.2 added a couple of new extensions (though one of them doesn't actually live in the Extensions menu), and included a large number of rewrites and bug fixes to existing extensions. If you're a frequent user of extensions, it might be worth checking the release notes to see if there have been changes to any that you rely on: https://wiki.inkscape.org/wiki/index.php/Release_notes/1.2#Extensions

The first of the new extensions will only be of interest to users who

import SVG files that were exported from Adobe Illustrator. Extensions > Document > Process Illustrator SVG… should be run immediately after importing such a file, in order to sort out issues with layer names and document scaling. If you are the one doing the exporting, it's also worth taking a look at this extension before you export the file from Illustrator; launching the extension displays this dialog, which offers tips on how best to export in the first place.



The Clipart Importer extension is a replacement for the old 'File > Import Clip Art…' tool that was removed from Inkscape a few versions ago. Because of this, it doesn't appear in the Extensions menu, but rather as File > Import Web Image. It allows you to search several different sources of clip art images for files that you can then import directly into your document. Note that you have to hit Enter in the search field to trigger the search, and if you switch sources you'll have to focus the search field and press Enter again.

Compared with the old tool, this one has a big advantage in terms of displaying licensing information where it's available, helping to ensure that your own creations stay on the right side of copyright law.

In my experience the dialog doesn't like being un-focused – which is a problem if, like me, you use a focus-follows-mouse configuration on your machine. It can lead to the appearance of Inkscape being unresponsive, and even clicking on the window didn't re-focus it for me. Using Alt-Tab to bring the dialog back to the top of the stack did the job, so it's probably worth trying that before reaching for the kill command. You should probably save your work as a precautionary measure before opening this dialog, just in case.

And that's it for the 1.2 series. There are still some smaller features and changes that I haven't discussed, but I think I've covered all the really big and important things. Inkscape keeps pressing on with new releases, so my focus is now on more recent releases. See you next month for more new features that landed with 1.3.x.

Mark uses Inkscape to create comics for the web (www.peppertop.com/) as well as for print. You can follow him on Twitter for more comic and Inkscape content: **@PeppertopComics**

Inkscape - Part 142

This month, I'll begin with a small correction, courtesy of @dwhall on the Inkscape forum (https://inkscape.org/forums/). Previously I described the new ability to set the origin point for Selector tool transformations – i.e. the point that remains fixed when you change the width or height via the numeric fields in the tool control bar. I said that it was a shame that there's no way to choose the center-center point as the reference. It turns out there is, but not while the Selector tool is in its default move/resize mode. Instead, you have to click on the object a second time to switch to the scale/rotate handles. Once in that mode, a single click on the handle that marks the center of rotation will display the blue color and barely visible lines that indicate it's being used as the reference point.

This also means that the reference point can be moved to an arbitrary position by dragging the handle (Shift-Click on it to return it to the center of the bounding box), but suffers from a long-standing omission in this regard in that there's no way to set the position by typing values in, except via the not-so-user-friendly XML editor. In addition, none of the other rotate/skew handles can be clicked to set the reference point, so if you want to switch it to one of the corners, for example, you'll need to cycle back round to the move/resize mode.

I sincerely hope that the Inkscape developers make these three small additions in a future release to unify this new functionality across the different modes of the Selector tool:

- Add a visible center handle to the move/resize mode, similar to the center of rotation
- Add fields for explicitly entering the reference coordinates to the tool control bar
- Allow all the scale/rotate handles to be selected as reference points

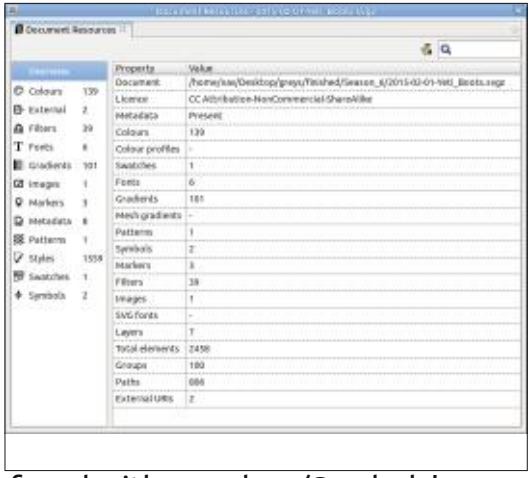
Document Resources Dialog

This dialog is a new addition to version 1.3, and provides a single place to view the various 'assets' that are used in your document. An asset, in this case, is one of the following types of element:

- Fonts
- Styles
- Colors
- Swatches
- Gradients
- Filters
- Patterns
- Symbols
- Markers
- Images
- External references (i.e. links to URLs)
- Metadata

There's an absolute trove of information in this dialog, as can be seen from the default 'Overview' pane that appears when it's first opened via File > Document Resources.

To the left of the dialog you can see a list of the various categories of information available, beginning with the Overview pane that is currently selected. Categories appear in this list only if they have content to display, so you won't be



faced with a useless 'Symbols' category if your document doesn't contain any symbols, for example.

To the right of this list is the main content of the dialog, which often displays information in a tabular form, as can be seen with the Overview. Unfortunately, the cells of these tables are not only read-only, but also don't allow their contents to be copied to the clipboard. There's also no option to export them to a file for further processing or analysis. A command-line mechanism for opening a drawing in a headless Inkscape instance and exporting a JSON file from this dialog would open up a lot of possibilities for indexing or

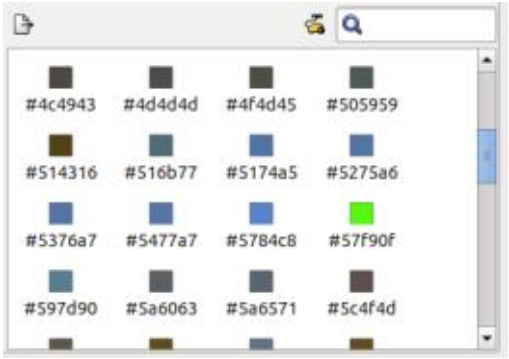
exploring any large corpus of SVG files.

Looking at the overview pane, you can see some immediately useful information about the document path, the licence and the total number of elements. A lot of the other information, however, just duplicates the list of categories on the left. There's no interactivity in the overview table, either: clicking or double-clicking on one of those entries is in vain, as it doesn't take you to the relevant category pane, and certainly doesn't open the appropriate dialog on the screen. For example, it would be great on noticing that I've failed to add a licence, to be able to double-click on the second row as a shortcut to open the Document Properties dialog with the relevant tab selected, but this dialog doesn't currently provide that sort of capability.

Above the main pane is a toolbar whose contents change depending on the selected pane. In all cases the button and search box on the right remain present. The button – which has an icon that looks like a vacuum cleaner, in my theme – will remove any definitions that are present in the file, but not used in the actual drawing. As an example, this can happen when you create gradients on objects then change them to a different fill, or delete the objects entirely. If that sounds familiar, it's because there is already a File > Clean Up Document menu item which performs the same task. In fact, if you have the 'Show icons in menus' option enabled in the Theming pane of the Preferences dialog, you'll probably already recognise the icon in this button.

The search box in the toolbar is actually more of a filter, in that it just restricts the entries in the current pane to those that match the search string. It doesn't search across all the panes in the dialog. This is a simple case-insensitive substring search that doesn't support any wildcards, which somewhat limits its usefulness (e.g. no searching for '??ff??' in the Colors pane to find all the colors with 100% green).

I won't go through every pane in this dialog, as many show just tabular information. But I will take a look at some that do offer some additional functionality, starting with the aforementioned Colors pane.



This pane shows a grid of color swatches, each with its hexadecimal RGB value beneath it. Remember, no matter what color picker you use, Inkscape stores values internally as RGB, so don't go expecting to see your HSL values in here. Also note that these are 6-digit hex values, so any transparency in your colors is omitted. This is a pity, because the difference between a fully opaque and a nearly transparent color is stark, so this view might grossly misrepresent the colors that are actually visible in your image.

You can't edit the hex values from here; you can't double-click to open a color picker to change any of the colors interactively; you can't even drag a swatch to the canvas to set the fill or stroke on your selected object. This is purely a report of the colors used in your image, in ascending hex order, not a tool for working with them.

There is, however, one small bit of additional functionality in this pane, offered by the button in the toolbar. Clicking that will export a palette of the colors in *.gpl format – the palette format used by The Gimp and some other tools, as well as by Inkscape itself. But before you get too excited about this function, note that it will always export the entire list of colors. There's no way to select a subset of the swatches in the pane to produce a more limited palette.

The Fonts pane is a useful little tool, showing a list of the fonts used in your document, but doing so with two entries per font: the first shows the font name rendered at a fairly large size using the font itself, while the second shows it as plain text.

This is great for getting an overview of the fonts you're using, and the preview makes it easier to identify whether each font is a serif, sans-serif, display font, and so on. Showing the font name as plain text also helps when dealing with dingbat fonts, or extremely fancy or cursive fonts where the fully rendered version might be hard to read.

Unfortunately color fonts don't appear in color (e.g. Gilbert Color in the screenshot). Even if there are technical reasons why they can't be rendered in color in here, it would be nice to at least have some flag or indicator to show that they're color fonts. In fact, a little more metadata about the fonts would be nice to have in here generally.

The Gradients, Markers, Patterns, Swatches and Symbols panes all share some common toolbar buttons, as seen here in the Markers pane.

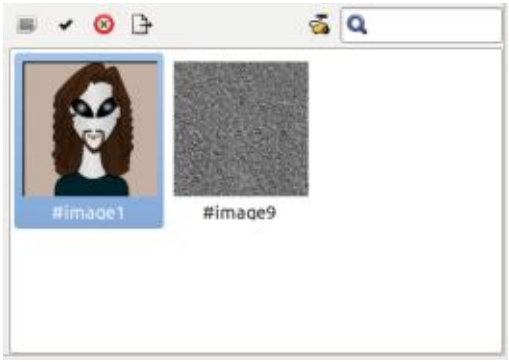
Unfortunately there only seems to be a symbolic icon for the first button – at least in my 1.3.2 AppImage downloaded directly from the Inkscape website. All other themes show some sort of 'missing image' icon. For reference, the symbolic icon for this is a pencil.

The buttons are enabled only when an item in the pane is selected. Clicking the first will let you edit the label of the item; double-clicking on the item label directly also achieves the same result. I'll make my usual clarification that the label is purely an informative thing that Inkscape offers. It's not part of the SVG spec (it's stored as an attribute in the Inkscape namespace), and changing this won't alter the ID, which makes it less useful for JavaScript developers. But if you want some of your assets to have user-friendly labels – perhaps to make them easier to reuse – you can rename them in bulk (albeit one at a time) in this dialog.

The tooltip for the second button in the toolbar claims that it will "Select this on canvas (if applicable) or in the XML dialog (e.g. a pattern in the 'defs' section)". In my experience, this should probably be reworded to be the other way round: it will actually select the item in the XML dialog, if it's already open, and very occasionally also select an element on the canvas. The reason for this is that most of the items in these panes refer to internal definitions that don't have a direct on-canvas object associated with them. Selecting the definition in the XML editor makes sense in that case – though it's not something that's likely to be terribly useful for most people. Only if the item is actually represented directly on-canvas will the corresponding object be selected. This will not be the case for any of the panes I listed above though it may be for images, which I'll discuss further below.

The last button on the toolbar deletes the selected item from the file. This will not always have the effect you expect. Deleting a Swatch, for example, will remove the fill or stroke color from any objects that use it – no surprise there. But deleting a Symbol won't remove it from your canvas; instead the symbol is converted to a group of normal objects, the same as if you'd selected it on the canvas and used Edit > Clone > Unlink Clone.

There's one more pane that features all three of these toolbar buttons, plus one additional entry: the Images pane.



The extra button here lets you directly export the selected image to disk. This is the same functionality that is present in the Image Properties dialog on the right-click context menu for images, but it's useful to have it in the Document Resources dialog too, particularly if you have to extract multiple images from a single document.

The remaining panes – for Metadata and Styles – just show data in a tabular form, with no additional tools or features. Again, this is rather frustrating. The lack of

even a means to copy the text to the clipboard seems odd, and trying to show long CSS definitions in a single row in a table isn't terribly usable. At least the search field can be used to narrow down the list of styles, if you're looking for something specific. But the lack of wildcards makes even this tricky. Perhaps the developers should reuse the pop-up editor for styles that is present in the XML editor, which at least breaks the long string down into its constituent parts.

It may seem as though I've done little but complain in this article, but the truth is that I really like the idea of a Document Resources dialog. I just wish that the implementation went a little further, with a few more convenience features that would make this more of a go-to dialog for working with resources in bulk. As it stands, it's mostly a reporting tool that doesn't actually produce reports, rather than something I think I'll be using regularly.

But this is the first appearance of this dialog in Inkscape, and I can only hope that the developers will iterate on it over time, adding more functionality. It also seems like it might be a useful target for new Inkscape developers, as there's scope for some simple and isolated additions (such as the ability to copy text from the tables) which would probably make a good choice for someone's first contribution to the project.

Although this is the only brand-new dialog in version 1.3, several others have seen changes and improvements which I'll look at next month.

Mark uses Inkscape to create comics for the web (www.peppertop.com/) as well as for print. You can follow him on Twitter for more comic and Inkscape content: **@PeppertopComics**