RFC 7996 : SVG Drawings for RFCs: SVG 1.2 RFC

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 16 décembre 2016

Date de publication du RFC : Décembre 2016

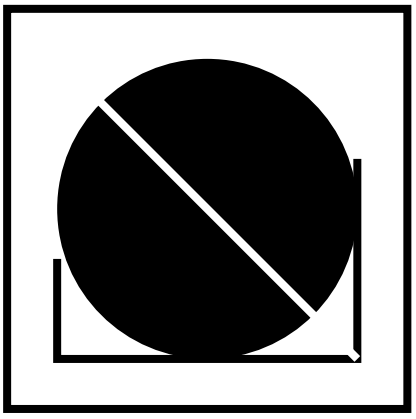
https://www.bortzmeyer.org/7996.html

Dans la longue liste de RFC décrivant le nouveau format des RFC (commencez par le RFC 7990 [1]), ce document décrit l'utilisation de SVG pour faire (enfin) des graphiques dans les RFC.

Traditionnellement, en effet, seul le texte était possible dans les RFC. (On pouvait toutefois faire des graphiques en art ASCII, dont on imagine bien que ce n'était ni très pratique à écrire - malgré l'excellent mode Emacs `picture-mode`, ni très facile à lire.) Il y avait de bonnes raisons à cet état de fait, notamment le manque d'un format d'images ouvert et largement répandu. Je parle bien sûr d'images vectorielles car, a priori, dans un RFC, il y aura beaucoup plus de schémas que de photos.

Le processus de décision a été long et compliqué. En 2013, le RFC 6949 notait déjà la décision de permettre des images, en indiquant « *"Graphics may include ASCII art and a more complex form to be defined, such as SVG line art"* ». C'est désormais possible. Au fait, c'est quoi, SVG ? Il s'agit d'une norme d'un format d'images vectoriel, gérée par le W3C, et fondée sur XML. Voici un exemple d'un simple dessin en SVG :

```
<svg xmlns="http://www.w3.org/2000/svg" version="1.2">
  <rect x="25" y="25" width="200" height="200" fill="white" stroke-width="4" stroke="black" />
  <circle cx="125" cy="125" r="75" fill="black" />
  <polyline points="50,150 50,200 200,200 200,100" stroke="black" stroke-width="4" fill="none" />
  <line x1="50" y1="50" x2="200" y2="200" stroke="white" stroke-width="4" />
</svg>
```



Et voici son rendu :

Les RFC n'utiliseront qu'un sous-ensemble de SVG, spécifié ici. Il existe d'autres sous-ensembles de SVG comme SVG Tiny, prévu pour des équipements contraints, genre "*smartphones*". Ce SVG Tiny a servi de base au SVG des RFC, sous-ensemble limité car on n'attend pas de dessins artistiques et compliqués dans les RFC.

SVG RFC est donc SVG Tiny **moins** les éléments permettant le multimédia et l'animation (pas de vidéos dans les RFC), l'interaction (pas d'interface utilisateur active dans un RFC), l'utilisation de langages de programmation (pas de JavaScript actif dans un RFC). Plusieurs autres restrictions ont été apportées : pas de couleur (RFC 6949, section 3.2), pas d'IRI, seulement des URI, et pas de choix arbitraire de police.

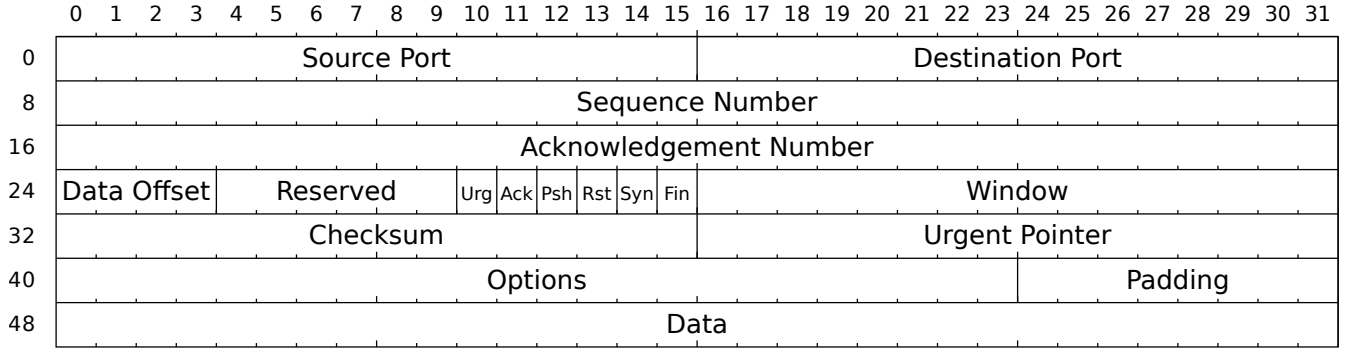
Comment on écrit du SVG ? S'il est évidemment possible de le faire entièrement à la main avec un éditeur ordinaire, gageons que peu de gens le tenteront. Notre RFC cite des éditeurs graphiques, produisant du SVG, comme les logiciels libres Inkscape et Dia. (Et, si on aime programmer en Python, il y a svgwrite <https://pypi.python.org/pypi/svgwrite>, que je présente plus en détail à la fin.) Attention, Inkscape et Dia produisent du SVG généraliste, pas du SVG RFC, qui est plus restreint. (Je ne connais personnellement pas d'outil pour produire du SVG RFC, ou pour « réduire » un fichier SVG généraliste en enlevant tout ce qui n'appartient pas à SVG RFC. Un tel outil était prévu mais je ne sais pas où il en est. C'était une des fonctions attendues du futur svgcheck <https://iaoc.ietf.org/documents/svgcheck-SOW-02.pdf>.)

Et l'accessibilité (section 4) ? Il est crucial que les RFC soient accessibles à tou[Caractère Unicode non montré [2]]te[Caractère Unicode non montré]s, non seulement que que soit le matériel utilisé, mais également quels que soient les handicaps dont souffre leur propriétaire. C'est bien joli de vouloir ajouter des tas de choses dans les RFC mais encore faut-il ne pas creuser ainsi davantage le fossé entre les utilisateurs. Ainsi, accepter de la couleur (le RFC 6949 limite les futurs RFC au noir et blanc) fait courir le risque que les daltoniens ne puissent pas comprendre un RFC. De même, les graphiques, qui ont l'air comme ça d'être une bonne idée, peuvent aggraver la situation des malvoyants. Le texte seul peut toujours être lu à voix haute par un synthétiseur de parole mais pas le graphique. Comme le note le RFC avec humour, « lire le source SVG à voix haute ne va pas le faire ».

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc7990.txt
2. Car trop difficile à faire afficher par LaTeX

Le texte « "*Tips for Creating Accessible SVG*" <http://www.sitepoint.com/tips-accessible-svg> » donne des bons conseils pour faire du SVG accessible. Et il y a bien sûr la norme ARIA, dont il existe une introduction <http://www.w3.org/TR/2010/WD-wai-aria-primer-20100916> et de bons exemples <http://blog.paciellogroup.com/2013/12/using-aria-enhance-svg-accessibility>. (Désolé, je n'ai pas suivi ces excellents principes dans les exemples ci-dessous, mais j'accepte les "*patches*".)

Si vous voulez voir des exemples concrets, regardez <https://www.cs.auckland.ac.nz/~nevil/SVG_RFC_1.2/>. Ainsi, l'exemple de schéma d'un en-tête TCP donnera : Et le schéma de la communi-



cation SIP du RFC 4321 fera

L'annexe A de notre RFC donne un schéma complet (formulé en Relax NG) du SVG des RFC. Il est extrait ici dans le fichier (en ligne sur https://www.bortzmeyer.org/files/svg-rfc.rnc), que vous pouvez utiliser pour tester la conformité de vos SVG. Par exemple, avec rnv <http://www.davidashen.net/rnv.html> :

```
% rnv files/svg-rfc.rnc files/tcp-header.svg
files/tcp-header.svg
```

En revanche, avec du SVG trop « riche » (ici, utilisant les couleurs), on aurait :

```
% rnv files/svg-rfc.rnc /tmp/test1.svg
...
/tmp/test1.svg:2:267: error: attribute ^fill with invalid value "red"
required:
value ^token "none"
value ^token "black"
value ^token "white"
value ^token "#000000"
value ^token "#FFFFFF"
value ^token "#ffffff"
value ^token "inherit"
```

Une alternative, pour tester la validité des SVG conformes à ce profil, sera svgcheck quand il sera développé <https://iaoc.ietf.org/documents/svgcheck-SOW-02.pdf>.

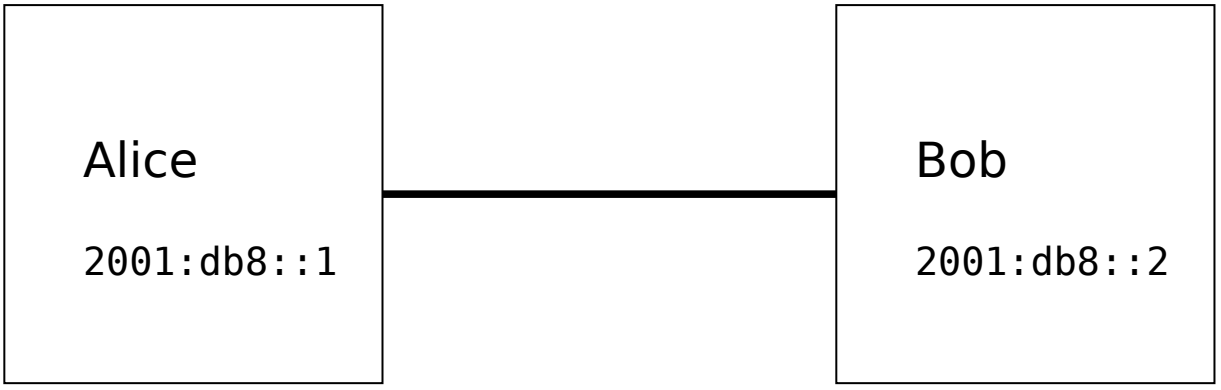
Avec Inkscape, il faut veiller à sauver le fichier en "*Plain SVG*" (autrement, on a des ennuis avec les éléments spécifiques d'Inkscape, ex-Sodipodi). Mais il reste malgré cela deux ou trois trucs à corriger manuellement, avant que le document produit par Inkscape soit accepté. Pour Dia, il faut utiliser l'action

"*Export*" (par défaut, Dia n'enregistre pas en SVG), mais Dia produit alors un document avec une DTD. Si on la retire (manuellement, ou bien avec `xmllint --dropdtd`), tout se passe bien, le document SVG est alors conforme au profil demandé pour les RFC.

Autre solution que j'aime bien pour faire du SVG, dès qu'on a des éléménts répétitifs et qu'on veut donc automatiser (en Python), svgwrite <https://pypi.python.org/pypi/svgwrite>. Ce schéma en art ASCII :

```
+--------------+      +----------------+
|  Alice       |------------------------------------|      Bob       |
| 2001:db8::1  |                                    |   2001:db8::2  |
+--------------+      +----------------+
```

aurait pu être créé avec svgwrite avec (en ligne sur `https://www.bortzmeyer.org/files/network-schema-svgwrite.py`), qui donne



Bien sûr, pour un schéma aussi simple, le gain n'est pas évident, mais il le devient pour les schémas comportant beaucoup d'éléments récurrents. Mais notez que svgwrite ne connait pas le profil « SVG pour RFC » et, par défaut, peut donc produire des SVG invalides (par exemple avec de la couleur). Le programmeur doit donc faire attention.