

# RFC 9896 : SVG in RFCs

Stéphane Bortzmeyer  
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 5 février 2026

Date de publication du RFC : Janvier 2026

<https://www.bortzmeyer.org/9896.html>

---

Le RFC 7996<sup>1</sup> avait introduit la possibilité de mettre des images dans les RFC. Ce nouveau document le remplace, il décrit les règles pour inclure du SVG. Il est très court, se limitant aux principes, il n'a plus de mention d'un profil SVG particulier.

On peut donc mettre des images vectorielles car, a priori, dans un RFC, il y aura beaucoup plus de schémas que de photos. Donc, pas de *"bitmap"*.

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 :

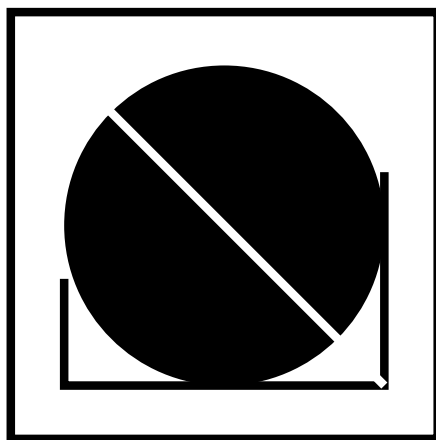
```
<svg xmlns="http://www.w3.org/2000/svg" version="1.2">
  <rect x="25" y="25" width="200" height="200" fill="white" stroke-width="4" stroke="black" />
  <circle cx="125" cy="125" r="75" fill="black" />
  <polyline points="50,150 50,200 200,200 200,100" stroke="black" stroke-width="4" fill="none" />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

Et voici son rendu :

Ce RFC décrit des objectifs, une politique. Contrairement à son prédécesseur, le RFC 7996, il ne décrit pas de profil de SVG, il fixe des buts, pas les moyens de les atteindre. Ces buts sont :

---

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc7996.txt>



- Le schéma en SVG ne doit jamais être la seule représentation. Le RFC doit être compréhensible sans les schémas.
- Le SVG ne doit pas inclure d'animations et ne doit pas être trop réactif (pas question de changer quand on tourne le téléphone).
- Le but étant d'être vu et compris par le public le plus large possible, il ne faut pas utiliser les fonctions de SVG qui ne sont pas largement mises en œuvre. Pas de pointeurs vers des ressources externes, pas de script exécutable (pas de JavaScript actif dans un RFC...), attention portée à l'accessibilité, en suivant WAI. Pour la même raison, il ne faut pas de dépendance vis-à-vis des couleurs (RFC 6949, section 3.2) et pas de choix arbitraire de police.

Il est crucial que les RFC soient accessibles à tou-te-s, non seulement quel 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.

La traduction de ces principes en conseils techniques concrets, et l'application de ces règles seront du ressort du "*RFC Production Center*". Il suivra peut-être les nombreux détails pratiques décrits dans le RFC 7996 mais il n'y est pas obligé.

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. Le RFC 7996 citait 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, qui peut inclure des fonctions de SVG qui ne suivent pas les principes de notre RFC.

Autre solution, utiliser la bibliothèque Fletcher <<https://www.bortzmeyer.org/fletcher.html>> de Typst <<https://www.bortzmeyer.org/typst.html>> :

```
% typst compile --format svg essai-fletcher.typ
```

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 <<https://www.w3.org/WAI/ARIA/apg/>> et de bons exemples <<http://blog.paciellogroup>.

[com/2013/12/using-aria-enhance-svg-accessibility](https://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".)

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.

Si vous voulez voir des exemples concrets de RFC utilisant SVG, vous avez entre beaucoup d'autres, le RFC 8899, le RFC 9869 ou encore le RFC 9750. Ainsi, le RFC 8899 inclut un schéma qui a deux formes alternatives, en art ASCII :

```
<artwork type="ascii-art" name="" alt="" align="left" pn="section-4.4-2.1.2">
Any additional
  headers          .--- MPS -----.
      |             |             |
      v             v             v
+-----+
| IP | ** | PL | protocol data |
+-----+

      &lt;----- PLPMTU -----&gt;;
&lt;----- PMTU -----&gt;;
</artwork>
```

Et en SVG, que vous pouvez admirer dans le rendu HTML du RFC <<https://www.rfc-editor.org/rfc/rfc8899.html>> et son rendu en PDF <<https://www.rfc-editor.org/rfc/rfc8899.pdf>>.

Une alternative, pour tester les SVG est `svgcheck` <<https://github.com/ietf-tools/svgcheck>>. Ici avec le vrai source du RFC 9750 :

```
% svgcheck rfc9750.xml
INFO: File conforms to SVG requirements.
```

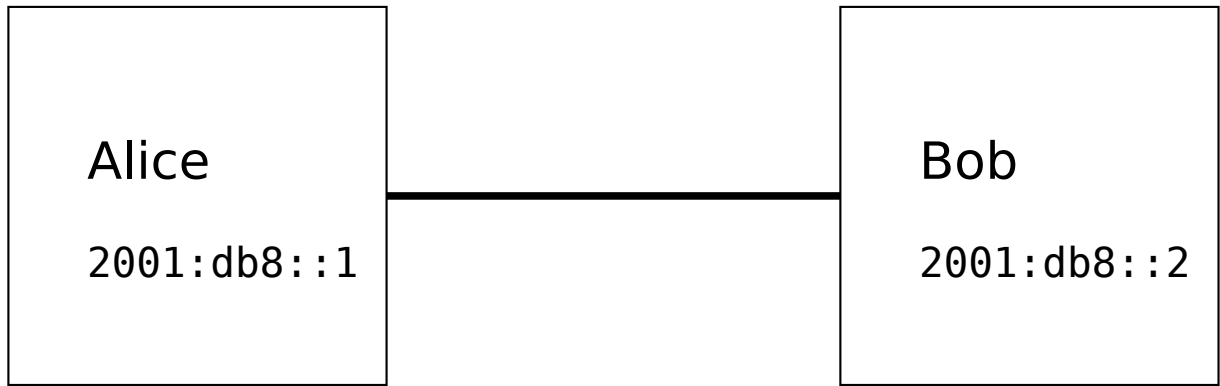
Et si je tente de mettre des couleurs vives en modifiant le XML :

```
% svgcheck rfc9750.xml
rfc9750.xml:438: The attribute 'stroke' does not allow the value 'red', replaced with 'black'
ERROR: File does not conform to SVG requirements
```

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

---

<https://www.bortzmeyer.org/9896.html>



```

+-----+ +-----+
| 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`, par défaut, peut donc produire des SVG non conformes aux règles du RFC (par exemple avec de la couleur). Le programmeur doit donc faire attention.