**Fichier de stéréolithographie**

Le **format de fichier STL** est un format utilisé dans les logiciels de [stéréolithographie](https://fr.wikipedia.org/wiki/St%C3%A9r%C3%A9olithographie) (STL pour **ST**ereo-**L**ithography). Ce format a été développé par la société 3D Systems. Ce format de fichier est utilisé par de nombreuses autres sociétés. Il est largement utilisé pour faire du [prototypage rapide](https://fr.wikipedia.org/wiki/Prototypage_rapide) et de la [fabrication assistée par ordinateur](https://fr.wikipedia.org/wiki/Fabrication_assist%C3%A9e_par_ordinateur). Le **format de fichier STL** ne décrit que la géométrie de surface d'un objet en 3 dimensions. Ce format ne comporte notamment pas d'informations concernant la couleur, la texture ou les autres paramètres habituels d'un modèle de [conception assistée par ordinateur](https://fr.wikipedia.org/wiki/Conception_assist%C3%A9e_par_ordinateur).

Le fichier STL décrit un objet par sa surface externe. Cette surface est nécessairement fermée et définie par une série de triangles (ou de facettes). Chaque triangle est défini par les coordonnées cartésiennes (x, y, z) dans un trièdre direct de son vecteur normal unitaire (n) orienté vers l'extérieur de l'objet et de ses trois sommets (vertex/vertices en anglais) ordonnés dans le sens trigonométrique. Pour certains logiciels, le vecteur normal peut être mis à (0,0,0) ; il sera alors calculé par le logiciel à partir des coordonnées des sommets suivant la règle de la main droite. Les coordonnées des sommets devaient à l'origine être positives, mais on rencontre désormais couramment des coordonnées négatives dans les fichiers STL. Chaque triangle doit partager 2 sommets avec chacun des triangles le juxtaposant. Dit autrement, le sommet d'un triangle ne doit pas être sur l'arête d'un autre triangle. Il n'y a pas d'information d'échelle. L'unité de longueur est arbitraire. Il est recommandé de classer les points par z croissant pour faciliter la lecture du fichier par certains logiciels. De plus, l'axe z est considéré comme l'axe vertical, bien qu'une correction sur l'axe y soit effectuée par certains logiciels de conception 3D comme [Blender](https://fr.wikipedia.org/wiki/Blender).

Il existe deux types de fichier : ASCII et binaire. Les fichiers ASCII sont faciles à créer, à la main ou avec n'importe quel langage de programmation, mais peuvent être très volumineux si l'objet est complexe. Les fichiers binaires moins volumineux sont plus utilisés pour cette raison. Les fichiers sont enregistrés avec l'extension .stl.

**ASCII STL**

Un fichier STL ASCII commence par la ligne :

solid *name*

où *name* est une chaîne facultative (mais si *name* est omis il doit toujours y avoir une espace après le code *solid*). Le fichier continue avec n'importe quel nombre de triangles, chacun représenté comme suit :

facet normal *ni nj nk*

outer loop

vertex *v*1*x* *v*1*y* *v*1*z*

vertex *v*2*x* *v*2*y* *v*2*z*

vertex *v*3*x* *v*3*y* *v*3*z*

endloop

endfacet

Où chaque *n* ou *v* est un [nombre à virgule flottante](https://fr.wikipedia.org/wiki/Virgule_flottante) au format « *signe*-[*mantisse*](https://fr.wikipedia.org/wiki/Mantisse)-e-*signe*-[*exposant*](https://fr.wikipedia.org/wiki/Exponentiation) », par exemple -2.648000e-002.

Le fichier est clôturé par :

endsolid *name*

La structure du format laisse à penser qu'il existe d'autres possibilités, (par exemple, les **facet normal** avec plus d'une boucle (outer loop) ou, de boucles avec plus de trois sommets. Dans la pratique, toutes les **facet normal** sont de simples triangles.

Les blancs (espaces, tabulations, retours à la ligne) peuvent être utilisés partout dans le fichier sauf au sein des nombres, ou des mots. Les espaces entre les codes « *facet normal* », et, « *outer loop* » sont obligatoires (ce sont des mots réservés).

**STL binaire**

Cette section est vide, insuffisamment détaillée ou incomplète. [Votre aide](https://fr.wikipedia.org/wiki/Sp%C3%A9cial:EditPage/Fichier_de_st%C3%A9r%C3%A9olithographie) est la bienvenue ! [Comment faire ?](https://fr.wikipedia.org/wiki/Aide:Comment_modifier_une_page)

Un fichier STL binaire est architecturé de la façon suivante :

* Les 80 premiers octets sont un commentaire.
* Les 4 octets suivants forment un [entier](https://fr.wikipedia.org/wiki/Entier_(informatique)) codé sur 32 bits, qui représentent le nombre de triangles présents dans le fichier.
* Ensuite, chaque triangle est codé sur 50 octets, selon la décomposition suivante :
  + 3 fois 4 octets, chaque paquet de 4 octets représente un [nombre à virgule flottante](https://fr.wikipedia.org/wiki/Virgule_flottante) correspondant respectivement aux coordonnées (x, y, z) de la direction normale au triangle.
  + 3 paquets de 3 fois 4 octets, chaque groupe de 4 octets représente un [nombre à virgule flottante](https://fr.wikipedia.org/wiki/Virgule_flottante) correspondant respectivement aux coordonnées (x, y, z) de chacun des sommets du triangle.
  + Deux octets représentant un mot 16 bits de contrôle.

Les nombres en virgule flottante sont représentés conformément à la spécification [IEEE 754](https://fr.wikipedia.org/wiki/IEEE_754) et en mode [little-endian](https://fr.wikipedia.org/wiki/Endianess) utilisé sur Windows notamment.

UINT8[80] – en-tête

UINT32 – Nombre de triangles

foreach triangle

REAL32[3] – Vecteur normal

REAL32[3] – Sommet 1

REAL32[3] – Sommet 2

REAL32[3] – Sommet 3

UINT16 – Mot de contrôle

end

**Codage de la couleur en STL binaire**

Il existe au moins deux extensions officieuses du format STL binaire permettant de coder la couleur des facettes :

* Les logiciels [VisCAM](https://fr.wikipedia.org/w/index.php?title=VisCAM&action=edit&redlink=1) et [SolidView](https://fr.wikipedia.org/w/index.php?title=SolidView&action=edit&redlink=1) utilisent deux octets à la fin de chaque triangle pour coder 15 bits d'information RVB :
  + les bits 0 à 4 codent l'intensité du bleu (0 à 31),
  + les bits 5 à 9 codent l'intensité du vert (0 à 31),
  + les bits 10 à 14 codent l'intensité du rouge (0 à 31),
  + le bit 15 est à 1 si la couleur est définie, ou à 0 si la couleur n'est pas définie.
* le logiciel [Materialise Magics](https://fr.wikipedia.org/w/index.php?title=Materialise_Magics&action=edit&redlink=1) utilise un en-tête de 80 octets pour représenter la couleur de l'objet complet. Si une couleur est définie, alors l'en-tête définit la chaîne ASCII "COLOR=" suivie de 4 octets définissant l'intensité du rouge, vert, bleu et la transparence, dans un intervalle de 0 à 255. Cela définit la couleur de l'objet complet, mais chaque facette peut redéfinir sa propre couleur. Magics permet également de définir l'aspect d'un matériau et la réflexion de la surface. À la suite de la chaîne "COLOR="<rgba>, il peut y avoir la chaîne "MATERIAL=" suivie de trois couleurs (3×4 octets) : la première couleur correspond à la teinte de la réflexion diffuse, la deuxième à la réflexion spéculaire, et la troisième à la lumière ambiante. Ce codage du matériau est préféré au codage de la couleur de l'objet.

Le codage de la couleur d'une facette est donné par 2 octets :

* les bits 0 à 4 codent l'intensité du rouge (0 à 31) ;
* les bits 5 à 9 codent l'intensité du vert (0 à 31) ;
* les bits 10 à 14 codent l'intensité du bleu (0 à 31) ;
* le bit 15 est à 1 si la couleur est définie, ou à 0 si la couleur n'est pas définie.

L'ordre de définition rouge, vert et bleu est inversé dans les codages de ces deux familles de logiciels ce qui pose des problèmes de compatibilité des fichiers car un logiciel lisant ces deux fichiers ne peut les distinguer. Il n'est pas non plus possible de définir une facette avec une transparence propre.