

Dat2QP

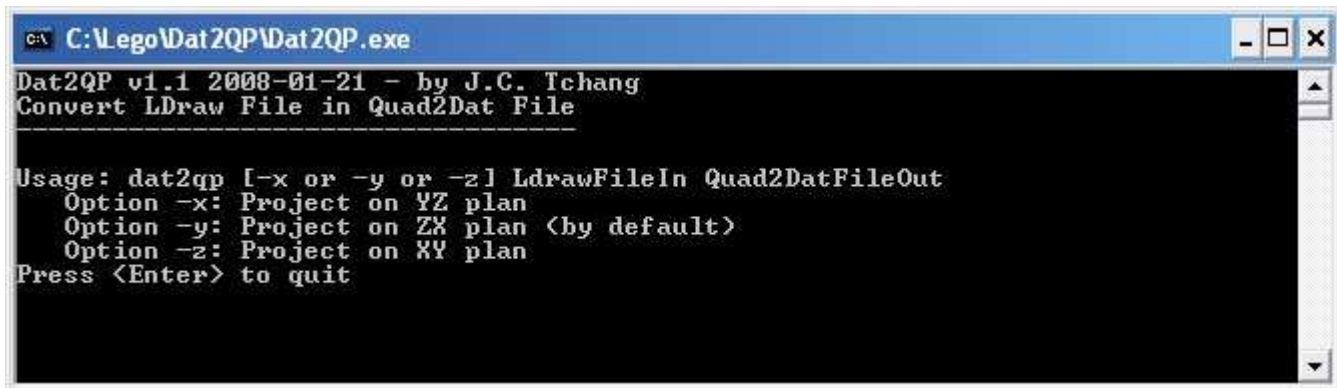
Mise à jour de la page : 12 mai 2008.



[J.C. Tchang](#)

[English Version
On Philo's Site.](#)

Dat2QP est un utilitaire écrit en C++ par J.C. Tchang. Il permet de convertir un fichier au format **LDraw** en fichier au format **Quad2Dat**.



```
C:\V\lego\Dat2QP\Dat2QP.exe
Dat2QP v1.1 2008-01-21 - by J.C. Tchang
Convert LDraw File in Quad2Dat File

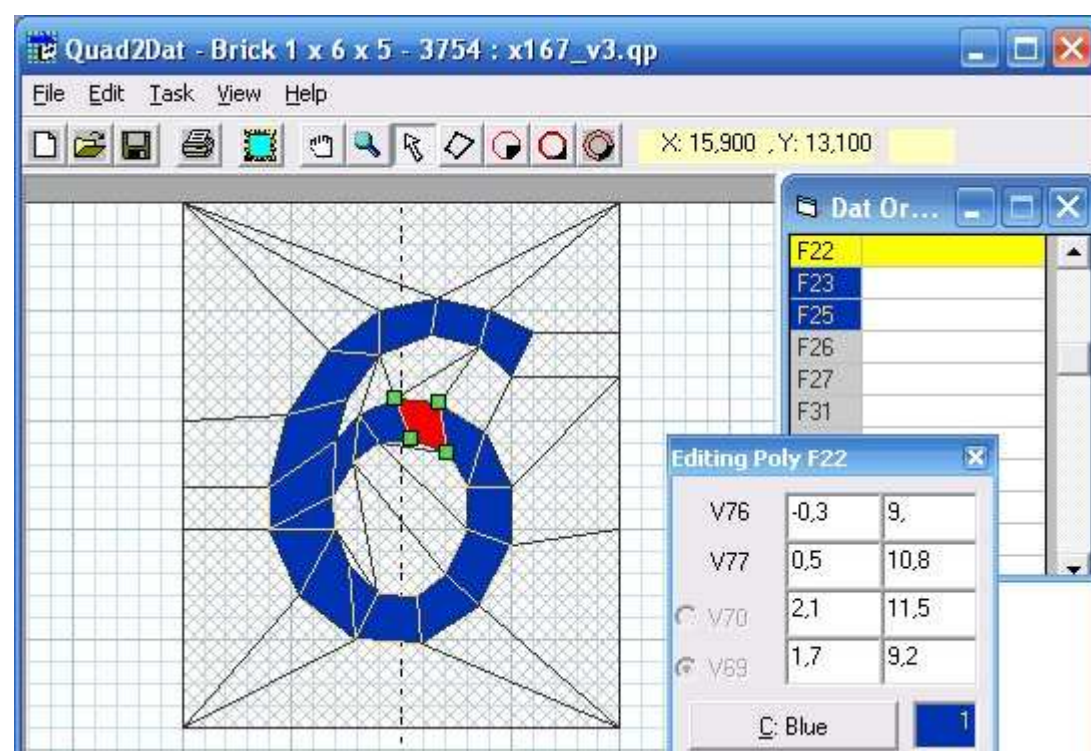
Usage: dat2qp [-x or -y or -z] LdrawFileIn Quad2DatFileOut
Option -x: Project on YZ plan
Option -y: Project on ZX plan (by default)
Option -z: Project on XY plan
Press <Enter> to quit
```

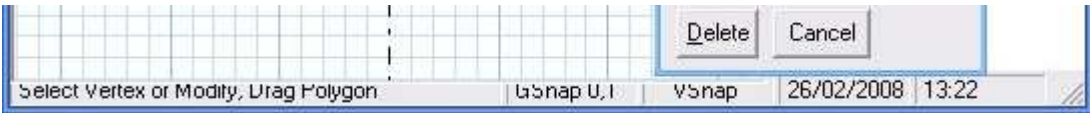
Exemple d'exécution de Dat2QP, donnant son usage.

Objet du programme

Quad2Dat est un programme de **Chris Alano**, permettant de créer des motifs pour des pièces LDraw. Ce programme, malgré ses imperfections (et plantages) reste le meilleur dans ce domaine. Merci Chris pour ce programme.

Vous pouvez le télécharger [ici](#), et accéder au [manuel en français](#).





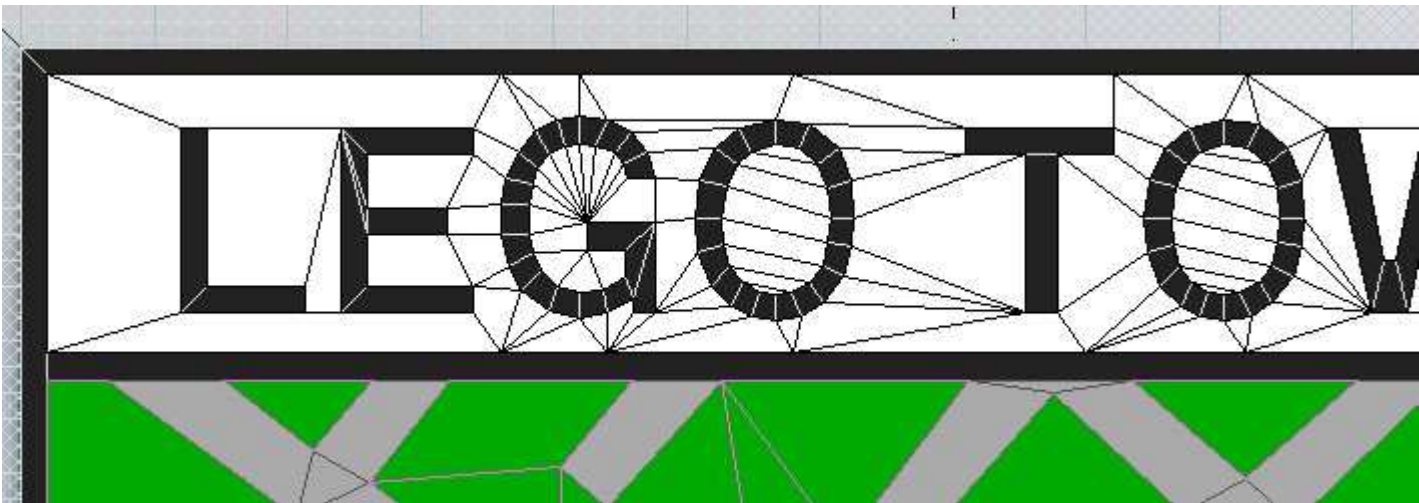
Exemple d'utilisation de Quad2Dat.

L'objet du programme **Dat2QP** est de permettre la récupération de motifs LDraw existants pour les modifier ou les compléter facilement avec **Quad2Dat**.

Dat2QP, convertit des polygones triangulaires et quadrangulaires 3D en 2D et écrit le résultat au format de fichier de Quad2Dat (fichier .QP). La direction de projection se fait suivant l'axe X ou Y (par défaut) ou Z.

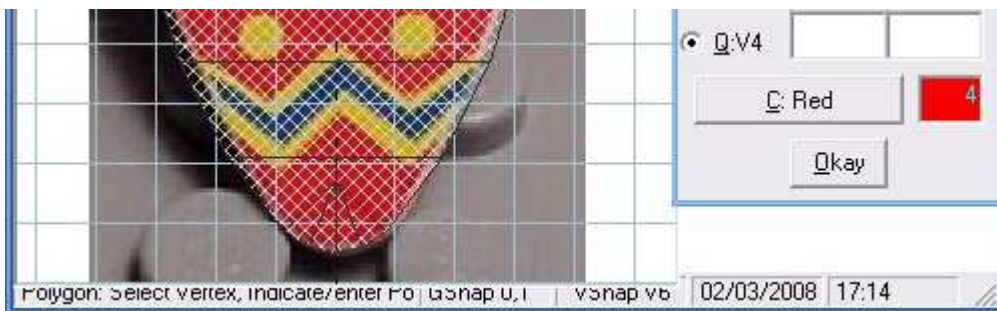
Un autre usage de **Dat2QP** est de pouvoir utiliser des outils extérieurs au programme **Quad2Dat**, comme :

- Dupliquer ou symétriser une partie répétitive avec **MLCad** ou autre programme.
- Utiliser **txt2Dat** pour créer une partie texte.
- Récupération de gabarit (Template) spécial.
- Etc ...



Exemple de duplication : La première lettre "O" a été créée dans **Quad2Dat**, puis la seconde a été dupliquée avec **MLCad** avant que le fond blanc entourant les lettres ne soit fait.





Exemple de gabarit (Template) spécial : La forme est récupérée du fichier LDraw 2586.DAT, après avoir décomposé (Inlined) les "disc" avec LDDP, et converti le résultat avec **Dat2QP**. Ici l'on voit la sélection d'un point sur le bord de la forme (V6).

Téléchargement

Vous pouvez [Télécharger ici](#) (902 Ko) le programme exécutable Windows avec son code source, et son manuel en Français et en Anglais (*Merci Philo*).

L'installation est des plus simples, puisqu'il suffit de décompresser le fichier et d'installer l'exécutable dans un répertoire.

Comme exemple, j'utilise le répertoire [c:\lego\dat2qp](#).

Utilisation

Création du fichier d'entrée

Créez un fichier d'entrée contenant les facettes à traiter, c'est-à-dire des polygones triangulaires (type 3) et/ou quadrangulaires (type 4). Le programme ne traite pas les primitives (type 1) qui doivent être décomposées (Inlined) par un programme tel que **LDraw Design Pad (LDDP)**.

Mettez ce fichier dans le répertoire d'installation de **Dat2QP**.

Conversion avec Dat2QP

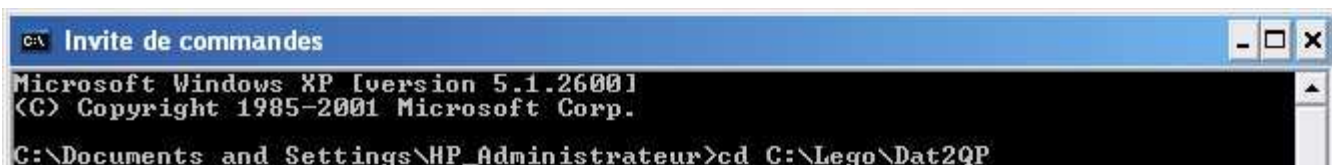
Ouvrez une fenêtre "Invite de commandes" (ex fenêtre DOS), puis allez dans le répertoire où est installé Dat2QP. Pour cela tapez la commande [cd c:\lego\dat2qp](#), ou bien si vous n'êtes pas un adepte des commandes DOS, tapez seulement [cd](#) suivi d'un espace, puis à partir de l'explorateur Windows positionné sur le répertoire d'installation de Dat2QP, faites glisser l'icône de dossier se trouvant dans la barre d'adresse, dans la fenêtre "Invite de commandes" puis sélectionnez cette fenêtre et validez avec la touche entrée. Cela est plus facile, surtout si le répertoire d'installation contient des noms longs.

Tapez la commande :

[dat2qp \[-x | -y | -z \] fichier_entree.dat fichier_sortie.qp](#)

exemple :

[dat2qp -y test1.dat test1.qp](#)




```

C:\Lego\Dat2QP>dat2qp -y test1.dat test1.qp
Dat2QP v1.1 2008-01-21 - by J.C. Tchang
Convert LDraw File in Quad2Dat File
=====
Option -y : Project on ZX plan
Input File : test1.dat
Output File : test1.qp

Read LDraw File:

Statistics:
0 Type 0 : Comment(s)
0 Type 1 : Primitive(s)
2 Type 2 : Line(s)
0 Type 3 : Triangle(s) (Agree Form)
4 Type 4 : Quad(s) (Agree Form)
0 Type 5 : Conditionnal Line(s)

Output:
4 Form(s)
16 Vectrice(s)

C:\Lego\Dat2QP>_

```

Exemple d'utilisation.

Utilisation du fichier de sortie avec Quad2Dat

Vous pouvez alors utiliser le fichier de sortie, [test1.qp](#) par exemple, avec **Quad2Dat**.

Nota : Si vous avez un message d'erreur à l'ouverture du fichier dans **Quad2Dat** du type "Run-time error '13': Type mismatch", c'est sans doute que vous êtes configuré avec un clavier "Français", utilisant la virgule comme séparateur de décimales. Il suffit de remplacer avec votre éditeur préféré tous les points "." par des virgules ",".

Le gabarit utilisé par défaut dans la conversion est la brique 1x6x5. Si les facettes converties avec Dat2QP ne rentrent pas dans le gabarit, il suffit de le cacher en allant dans le menu *View / Options...*, puis dans l'onglet "Display", décochez l'item "Template".

Voir le manuel de [Quad2Dat](#) pour plus d'informations sur ce programme.

Nota : Il est important de **SAUVEGARDER SOUVENT** sous des noms différents lorsque vous utilisez Quad2Dat, car dans de nombreux cas de figure ce programme plante avec perte des données précédentes.

Particularités du programme Dat2QP

Le programme **Dat2QP** vérifie la validité des polygones après projection, et élimine les formes dégénérées (sommets communs), mais pas les formes ayant 3 points alignés.

Par facilité, l'ordre des lignes dans le fichier généré est un ordre non trié. De même les sommets ayant les mêmes coordonnées sont gardés distincts.

Pour obtenir un fichier trié, avec suppression des sommets en doublon, **Quad2Dat** fait cela très bien lors du chargement. Il suffit donc de charger le fichier dans **Quad2Dat** et de le sauvegarder.

Le programme ne traite pas les primitives (disc, ndis, ring, ...) car en raison d'arrondis de calculs dans Quad2Dat les polygones jointifs d'une primitive n'ont pas les mêmes coordonnées.

Intégration de Dat2QP dans LDraw Design Pad

Il est possible de lancer **Dat2QP** à partir de LDraw Design Pad (Téléchargeable ici : [LDDP](#)) en créant 3 fichiers .BAT contenant :

Dans **dat2qp-x.bat** :

```
cd c:\lego\dat2qp
dat2qp -x %1 %2\dat2qp-x_result.qpp
```

Dans **dat2qp-y.bat** :

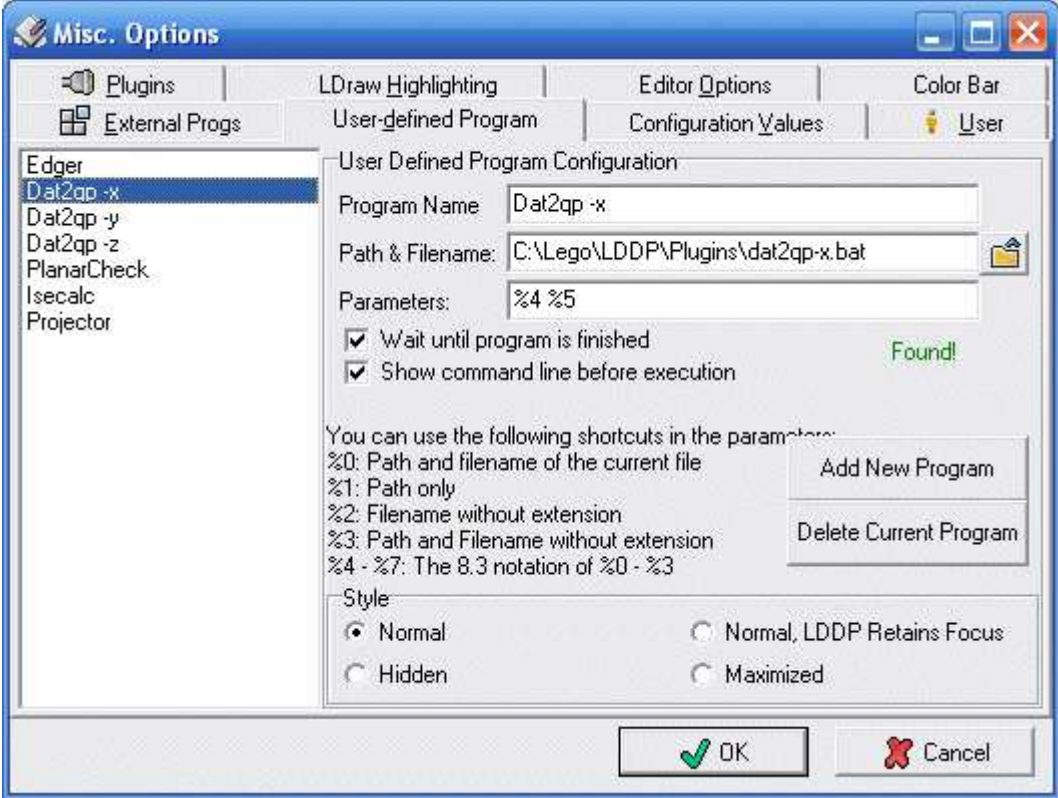
```
cd c:\lego\dat2qp
dat2qp -y %1 %2\dat2qp-y_result.qpp
```

Dans **dat2qp-z.bat** :

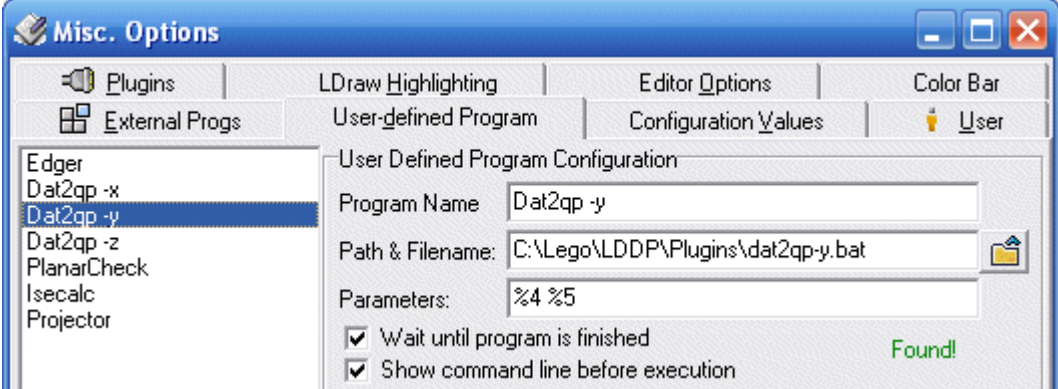
```
cd c:\lego\dat2qp
dat2qp -z %1 %2\dat2qp-z_result.qpp
```

Nota : Les fichiers doivent être placés dans le répertoire indiqué dans le champ "Path & Filename:" des boîtes de dialogue ci-dessous. Dans ces exemples, le répertoire des Plugins de LDDP.

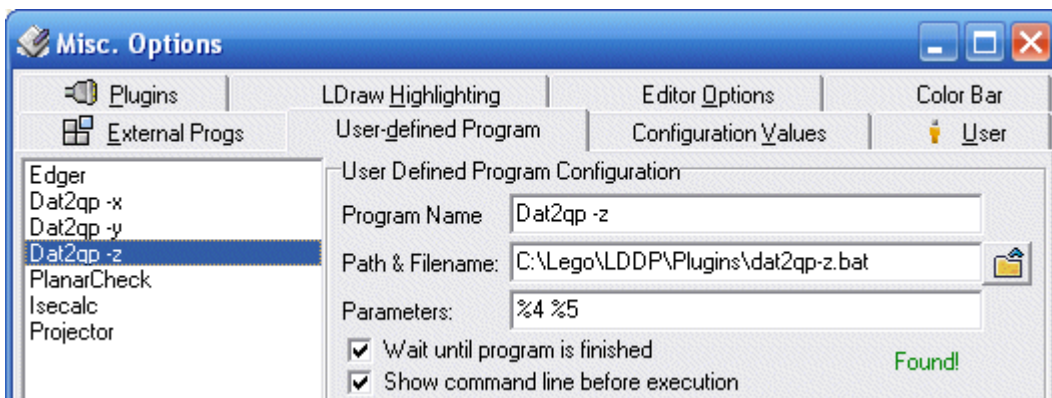
Puis de configurer LDDP comme suit :



Pour pouvoir traiter dans LDDP le fichier courant (en entier) suivant la direction X.



Pour pouvoir traiter dans LDDP le fichier courant (en entier) suivant la direction Y.



Pour pouvoir traiter dans LDDP le fichier courant (en entier) suivant la direction Z.

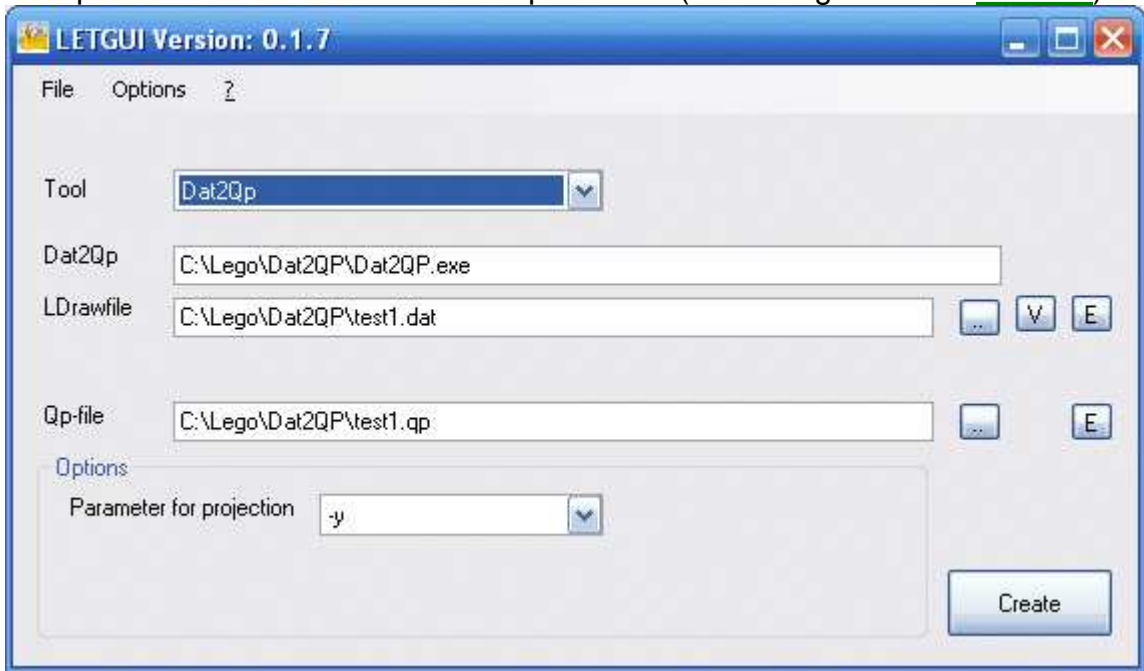
Une fois configuré vous pouvez lancer Dat2QP dans LDDP en allant dans le menu *Process/External Programs/User Defined Program*, ou en cliquant sur l'icône "User Defined Program" de la barre d'outils "External Programs" :



Le résultat du traitement se trouve dans le fichier [dat2qp-x_result.qp](#), ou [dat2qp-y_result.qp](#), ou [dat2qp-z_result.qp](#), placé dans le même répertoire que le fichier d'origine.

Intégration de Dat2QP dans LETGUI

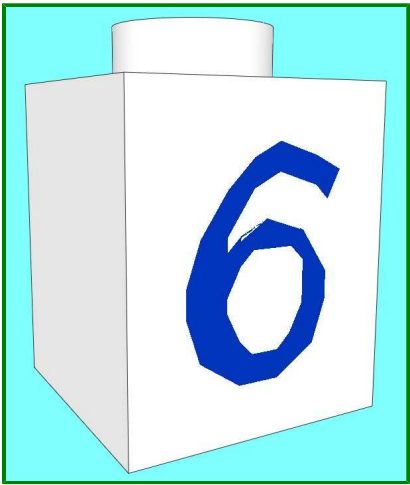
Il est possible de lancer **Dat2QP** à partir d'une interface plus conviviale que la ligne de commande. **LETGUI** créée par Michael Heidemann est fait pour cela. (Téléchargeable ici : [LETGUI](#)).



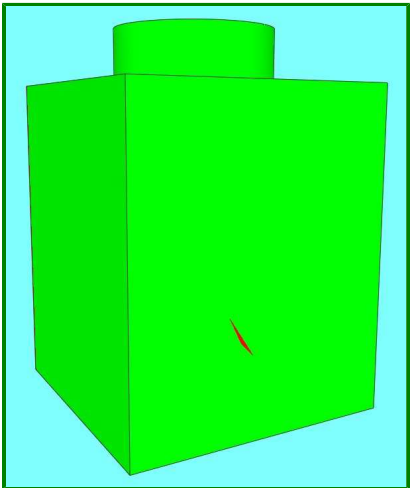
Tutoriel d'un cas réel

En cherchant dans la bibliothèque des pièces LDraw la brique 1x1 avec l'inscription 6, je me rends compte qu'il existe 2 pièces de ce type, dont la pièce 3005-6.DAT qui correspond à mon besoin.

En examinant cette pièce avec [LDView](#), je m'aperçois qu'elle n'est pas parfaite. Des facettes se chevauchent, et il y a un trou. Je décide de la reprendre pour l'améliorer.



(Cliquez sur l'image de gauche pour l'agrandir).

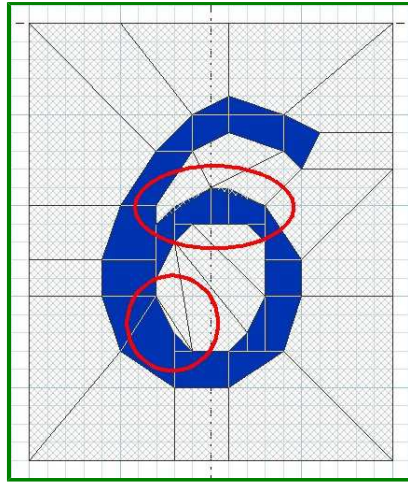
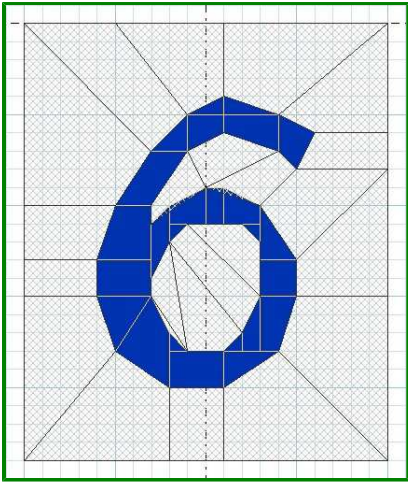


Le test de compliance BFC sous LDView montre la facette manquante.
(Cliquez sur l'image pour l'agrandir).

Dans un premier temps, je remplace toutes les faces et tenons de la partie standard des briques 1x1 par la sous-pièce s/3005s01.dat ne comportant pas la face avant. C'est l'occasion de voir que la face inférieure n'était pas correcte, et que l'utilisation de sous-éléments standards limite grandement ce genre de problème.

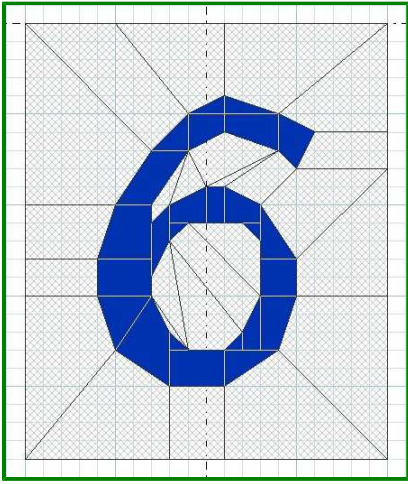
Ensuite, je convertis mon fichier au format Quad2dat en utilisant l'option de projection -z.

Un premier examen sous Quad2dat montre que 2 zones se chevauchent, et qu'il manque une facette triangulaire.



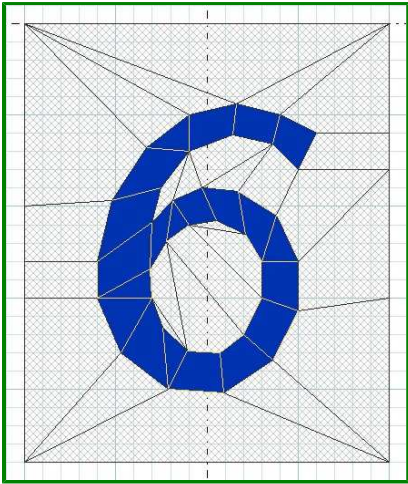
(Cliquez sur les images pour les agrandir).

Je supprime les facettes erronées, et refait les parties manquantes en y insérant des quadrilatères et triangles de la bonne couleur.



(Cliquez sur l'image pour l'agrandir).

Maintenant, il est possible d'aller plus loin, en supprimant les jonctions de facettes en "T", toujours avec la même méthode de suppression et création de facettes. Une sauvegarde, et rechargement du fichier peut-être alors utile pour supprimer les sommets en double. Ensuite, il est alors possible de déplacer certains sommets pour "arrondir" les courbes de la lettre, sans nécessairement ajouter des facettes.

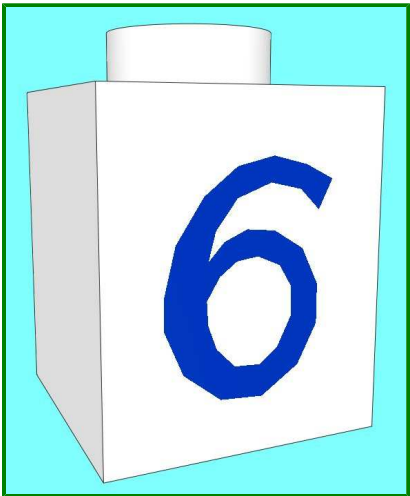


(Cliquez sur l'image de gauche pour l'agrandir).

Cela n'est pas parfait, mais cela ressemble un peu plus à la photo de droite. Je n'ai pas cherché à reproduire

exactement la hauteur du caractère, car il existe plusieurs variantes de polices utilisées sur cette pièce au cours des années.

Après exportation au format LDraw, et récupération dans le fichier de la pièce complète, nous obtenons notre pièce mise à jour.



(Cliquez sur l'image pour l'agrandir).

Droits et Copyright

Le programme **Dat2QP** (c) J.C. Tchang - 2008

- Version 1.1 : Ajout sélection direction de projection, ajout de tests, et suppression des polygones non valides.
- Version 1.0 : Version initiale.

Manuel en Français : J.C. Tchang.

Libre de droits pour tout usage non commercial. :-)
Les autres programmes cités dans cette page appartiennent à leurs auteurs respectifs.

Manuel en Anglais : Traduction [Philo](#).