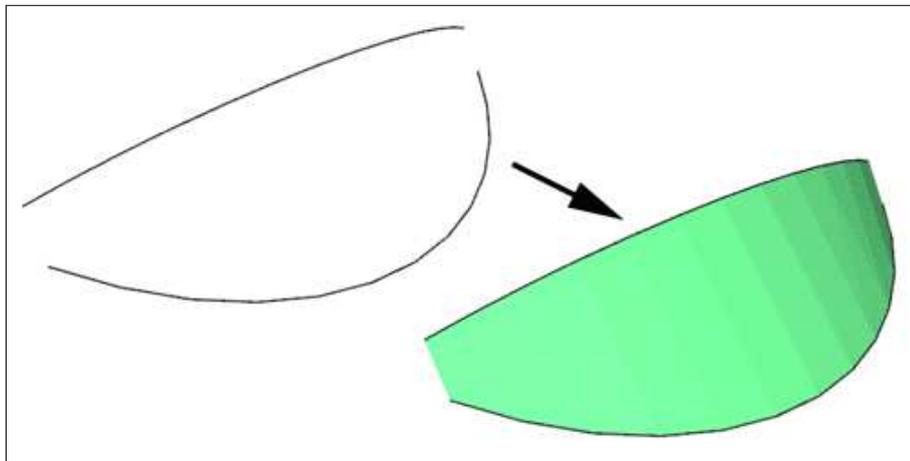


# Coverer, générateur de faces LDraw

Mise à jour de la page : 14 mars 2008.



**Coverer** est un utilitaire créant des faces composées de facettes triangulaires et quadrangulaires entre deux chemins formés de lignes. Chaque chemin doit être fourni à l'utilitaire dans un fichier LDraw séparé. Un troisième fichier contenant les facettes, et les lignes conditionnelles générées, est créé.

C'est une simple procédure de commande, dont le code source est fourni avec le programme ci-dessous, pour celui qui veut l'intégrer dans une interface plus conviviale.

## Téléchargement

Sur le site de l'auteur : [Coverer package](#), incluant le programme pour Windows, la documentation, les fichiers source (Visual C++ 6.0), et des fichiers exemples.

## Historique

- V1.0: Version Initiale

## Utilisation

- Préparez les fichiers d'entrée LDraw. **Coverer** utilise uniquement des lignes dans ces fichiers (lignes type 2). Les autres types d'entités sont ignorés. Si vous voulez calculer une face entre deux bords composés de primitives, vous devez d'abord les décomposer en lignes de type 2 (Inlined). [LDDesignPad](#) fait cela très facilement.
- Ouvrez une fenêtre "Invite de commande", et allez dans le répertoire d'installation de Coverer.
- Tapez la ligne de commande : `coverer [-r] [-l <length>] LdrawLineFile1 LdrawLineFile2 LdrawSurfaceFileOut`. **Coverer** crée alors le fichier `LdrawSurfaceFileOut` contenant les facettes. Notez, que si le fichier `LdrawSurfaceFileOut` existe déjà, il est écrasé sans message d'attention. `-r` et `-l` sont des paramètres optionnels contrôlant le comportement de **Coverer**. Ces options sont détaillées plus bas.
- **Coverer** écrit les valeurs dans le fichier de sortie avec 6 chiffres après la virgule. Cette précision est excessive pour la plupart des usages, et ces valeurs peuvent être arrondies. Ici encore [LDDesignPad](#) fait cela très bien.

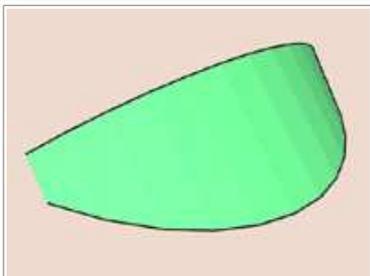
Voici une copie d'écran d'un exemple d'exécution du programme :

```
Invite de commandes
D:\Coverer>Coverer.exe arc1.dat arc2.dat arc-out.dat
Read Line File 1...
12 Lines in Line file 1
Read Line File 2...
12 Lines in Line file 2
Creating Triangles
24 Triangle(s) created
(0 degenerated suppressed)
Creating Conditional Lines
...and condensing coplanar Triangles into Quads
23 Conditional Line(s) created
2 Triangle(s) condensed into 1 Quad(s)
Writing output file
22 Triangle(s) written
1 Quad(s) written
23 Conditional Line(s) written
D:\Coverer>
```

## Comment Coverer fonctionne

- Les deux fichiers d'entrée sont lus et analysés. Tous les sommets de lignes sont mis dans des tableaux (limités à 2 000 lignes, ce qui devrait être suffisant !).
- Ces tableaux sont triés, permettant d'utiliser des fichiers LDraw dont les lignes ne sont pas ordonnées (par exemple des fichiers provenant d'[lsecalc](#)). Si plus d'un chemin est trouvé, un message d'erreur est affiché, car ceci dénote de mauvais fichiers d'entrée, et produirait un résultat erroné.
- Si l'option `-r` est présente, l'ordre du second fichier est inversé. Utilisez cette option lorsque la face résultante joint les chemins en inversant les extrémités (cela ressemble souvent à un papillon déchiqueté).
- Si l'option `-l xxx` est utilisée, les lignes de longueur supérieure à `xxx` sont coupées en segments plus petits. Vous pouvez obtenir des faces plus lisses en utilisant cette option.
- Les facettes triangulaires sont créées entre les deux chemins, progressant proportionnellement sur leurs longueurs. Lorsque des facettes dégénérées (réduites à une ligne) sont calculées (par exemple lorsque les chemins ont des points communs), elles ne sont pas créées.
- Une analyse des facettes triangulaires générées est faite. Si deux facettes consécutives sont dans le même plan et forment une facette quadrangulaire convexe, alors elles sont condensées. Si elles ne sont pas dans le même plan, une ligne conditionnelle est ajoutée (ligne de commande type 5).
- Le fichier de sortie est créé.

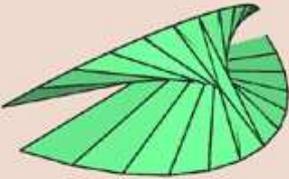
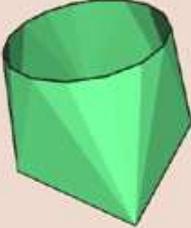
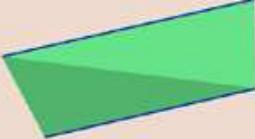
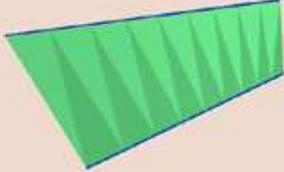
## Exemples



Une bonne face entre deux arcs.

Fichiers : arc1.dat, arc2.dat --> arc-out.dat

Ligne de commande : Coverer arc1.dat arc2.dat arc-out.dat

	<p>Ici, l'ordre et la direction des lignes du second fichier est inversé, créant cette forme étrange.</p> <p>Fichiers : arc1.dat, arc2-r.dat --&gt; arc-r-out.dat</p> <p>Ligne de commande : Coverer arc1.dat arc2-r.dat arc-r-out.dat</p> <p>Le paramètre <b>-r</b> corrige l'ordre, et génère la même face que dans le premier exemple.</p> <p>Fichiers : arc1.dat, arc2-r.dat --&gt; arc-rok-out.dat</p> <p>Ligne de commande corrigée : Coverer -r arc1.dat arc2-r.dat arc-rok-out</p>
	<p>Passer d'un carré à un cercle.</p> <p>Fichiers : sqc1.dat, sqc2.dat --&gt; sqc-out.dat</p> <p>Ligne de commande : Coverer sqc1.dat sqc2.dat sqc-out.dat</p>
	<p>Notez l'importance de l'orientation relative des deux chemins, même s'ils sont symétriques. Ici le cercle est tourné de 45°, le résultat est tordu.</p> <p>Fichiers : sqc1.dat, sqc2-45.dat --&gt; sqc-45-out.dat</p> <p>Ligne de commande : Coverer sqc1.dat sqc2-45.dat sqc-45-out.dat</p>
	<p>Cet exemple montre comment créer une face vrillée comme dans la pièce <a href="#">43721 slope wedge</a>. Deux uniques lignes forment les bords. Une utilisation simple de Coverer donne résultat grossier...</p> <p>Fichiers : wedge1.dat; wedge2.dat --&gt; wedge-out.dat</p> <p>Ligne de commande : Coverer wedge1.dat wedge2.dat wedge-out.dat</p>
	<p>...le paramètre <b>-l</b> améliore le résultat :</p> <p>Fichiers : wedge1.dat; wedge2.dat --&gt; wedge-l-out.dat</p> <p>Ligne de commande : Coverer -l 10 wedge1.dat wedge2.dat wedge-l-out.dat</p>
	<p>Un anneau particulier est probablement l'application la plus utile de Coverer. Cela permet de joindre des primitives standards à 16 ou 48 cotés, ayant des angles de départ différents. Je l'ai utilisé dans des pignons, roues et pneus, comme la pièce 30391 qui a une primitive de base 16, mais bordé de 40 bossages.</p> <p>Fichiers : cir21.dat, cir16.dat --&gt; adapt1621.dat</p> <p>Ligne de commande : Coverer cir16.dat cir21.dat adapt1621.dat</p>

## Droits et Copyright

Le programme Coverer et son manuel d'utilisation appartiennent à (c) Philo (Philippe Hurbain).

Vous pouvez le retrouver sur son site et sa [page des utilitaires LDraw](#), et sur la version d'origine en [anglais de cette page](#).

*Traduction et Adaptation : J.C. Tchang.*