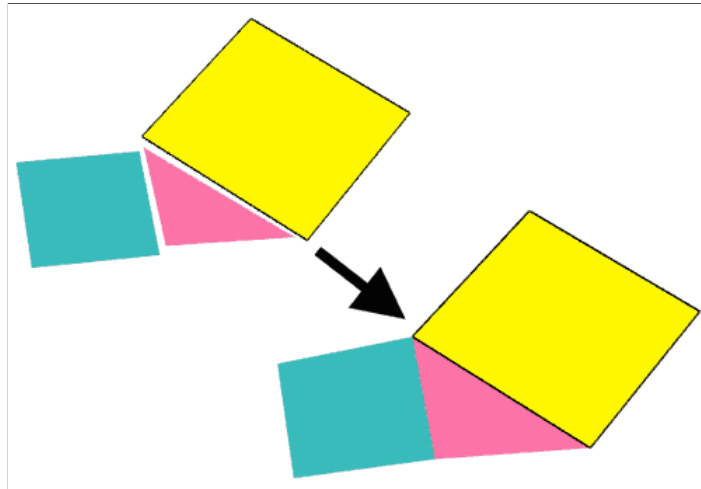


Unificator, a LDraw vertices unifier



Unificator utility allows to fine tune vertices coordinates in a LDraw file so that they match perfectly and close tiny gaps that may occur. This little tool derives from the vertices unification that I introduced in [SlicerPro](#). Vertices of simple elements (lines and surfaces) that are closer than some threshold are merged together at their center of mass. Another threshold controls the distance below which vertices snap onto fixed subparts/primitives vertices.

It is a simple console application, source code is provided below to anyone willing to integrate it in a more palatable user interface. You may also use Michael Heidemann [LETGUI](#) front-end (highly recommended!).

Download

[Unificator package](#), including program for Windows, documentation, source files (Visual C++ 6.0), sample files.

History

- V1.0: Initial release

Usage

- Input file maybe any valid LDraw file.
- Launch a command prompt
- Type the command line: Unificator [options] LdrawFile LdrawFileOut. **Unificator** will create LdrawFileOut, containing the file with unified vertices. Note that if file LdrawFileOut exists it will be overwritten without warning.
- **Warning:** unification process (especially with high threshold values) may warp some quads. You must check coplanarity on the result!
- **Unificator** outputs file with 6 digits after decimal point, this precision is excessive for most usages and values should be rounded. [LDDesignPad](#) or [DatHeader](#) does that very well.

Here is a screen shot of a sample run:

```
C:\WINNT\system32\cmd.exe
D:\UNIFIC~1>Unificator -ut 0.1 4496s01.dat 4496s01out.dat

Ldraw Unificator 1.0 - by Philo
-----
Vertices unification threshold: 0.1

Reading input file 4496s01.dat
249 lines in file
0 Edge lines
160 Conditional lines
0 Triangles
80 Quads
1 Subfiles

Unifying input file vertices

Writing output file 4496s01out.dat
249 lines written in file
0 Edge lines
160 Conditional lines
0 Triangles
80 Quads
1 Subfiles
Press <Enter> to quit
```

List of options

- -ut <val>: Vertices unification threshold (default 0.005) between line/surface vertices
- -us <val>: Vertices snapping distance (default 0.005) onto subpart/primitive vertices

- -s: snap onto subpart vertices
- -l "path": define ldraw search path (needed for primitive snapping)
- -p <val>: precision (default 0.001)

How Unificator works

- Input file is read and parsed. If -s is specified, subfiles are recursively inlined. Inlining occurs only with files in search path. For example to be able to snap on primitives, your LDraw folder path must be defined with -l option. All elements are stored in an array.
- Vertices of lines and surfaces are "unified". To do so, points close to each other are replaced by their center of mass. "Close" means their distance is lower than threshold defined by -ut parameter, 0.005 is the default value. This helps to close small gaps existing in the part.
- If -s option is specified, vertices of lines and surfaces are snapped onto nearby vertices of subfiles/primitives. "Nearby" means their distance is lower than threshold defined by -us parameter, 0.005 is the default value.
- Output file is written.

Examples



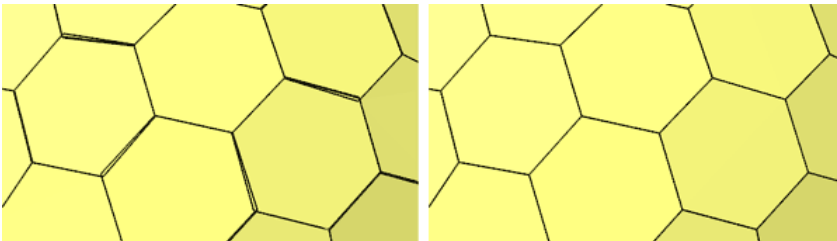
From left to right:

- The test input file, with a triangle, a quad and a rect primitive. There is a huge gap between these elements (about 11du wide).
- Result of **Unificator** with -ut 1 option. Triangle and quad vertices closer than 11du are merged together, but since -s is not specified, no snapping occur on the rect primitive
- The third image shows the result of unification with snap on subpart vertices. The 0.005 default unification threshold doesn't allow close points of quad and triangle to mesh together.
- With both options all nearby vertices merge together

Note that because the plane was not in the same plane as triangle and rect, the unification warps it and the resulting quad is no longer flat. You need to take care of this with Planarcheck or DatHeader. Of course, shape of triangle/quads are modified too. This is something to remember when using large threshold values.

Command line:

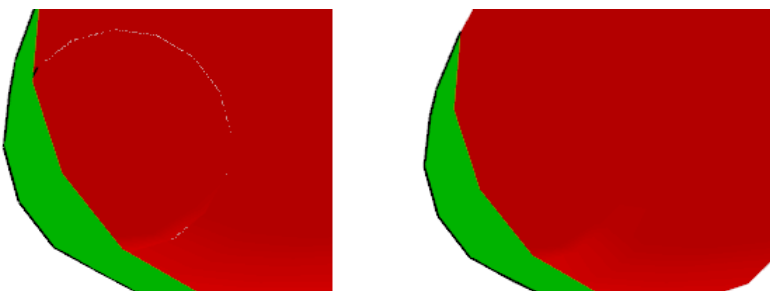
```
Unificator -ut 1 u1.dat ula.dat
Unificator -s -us 1 -l c:\ldraw u1.dat ulb.dat
Unificator -ut 1 -s -us 1 -l c:\ldraw u1.dat ulc.dat
```



A more practical case: initial version of 30208 faceted dome had a lot of inaccuracies. **Unificator** solves that issue easily (note that in the real case fixing the part was a bit more difficult because it is subparted in 6 sections. I initially used [SlicerPro](#) to repair the part, this induced me to create **Unificator**).

Command line:

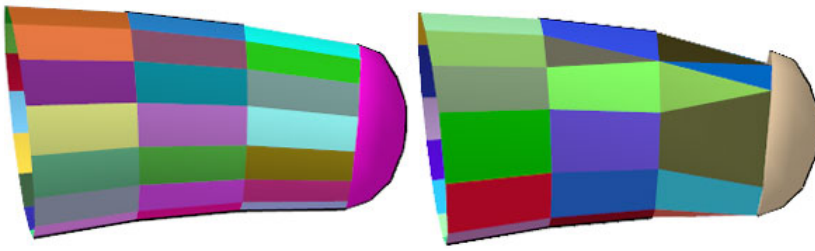
```
Unificator -ut 0.5 30208.dat 30208o.dat
```



Another (almost) real case example: the tip of 4496 pitchfork showed a tiny gap between tip sphere primitive and the rest of the tooth because of 3 digits rounding (the example file here was further rounded to 2 digits to make gap more visible). Snapping onto the sphere did the trick... Note that this method may be used to reduce the file size of parts that mix primitives and triangle meshes. First aggressively reduce precision, then run **Unificator** on the result to close the tiny gaps that appear between primitives and meshes.

Command line:

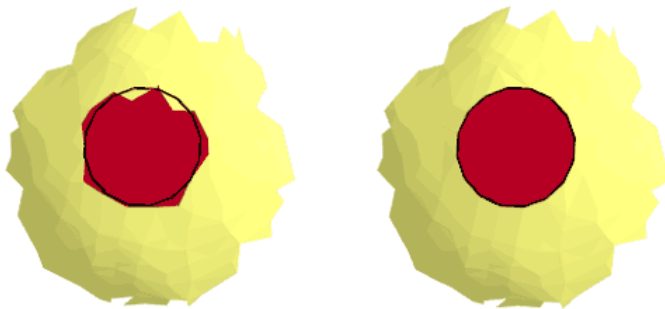
```
Unificator -s -us 0.03 -l c:\ldraw 4496s01.dat 4496s01o1.dat
```



Use carefully high unification threshold! Too high values result in badly distorted shapes... Since degenerated surfaces are automatically removed, conditional lines may be wrong after use of Unificator with high threshold values, in that case you have to process resulting file with Edger2.

Command line:

```
Unificator -ut 1 -s -us 1 -l c:\ldraw 4496s01.dat 4496s01o2.dat
```



Unificator may even be used to drill holes in a triangle mesh... Here we drop on the surface of the base of Fabuland cat a "burning circle" where we need to cut a hole for the neck. With a relatively high snapping threshold, vertices of the mesh are attracted onto circle vertices, resulting in a nice cut. Note that this trick will work properly only if the triangles are roughly the same size (or smaller) as circle segments. Everything must also be roughly in the same plane. You then only have to remove triangles inside the circle...

Command line:

```
Unificator -s -us 3 -l c:\ldraw u9108.dat u9108o.dat
```

