

# LDdraw.org

Centralized LDraw Resources

[Download](#) | [Reference](#) | [Community](#) | [Parts Library](#)

[Parts List](#) | [Activity](#) | [Submit](#) | [PT Tools](#) | [PT Reference](#)

[Library](#): [Parts Tracker](#): [Reference](#): Primitives

## LDdraw Primitives Reference

This page is a source of reference for the LDraw primitives in the \LDdraw\p directory. Primitives are defined as highly re-usable components of LEGO parts modelled for LDraw.

They serve several purposes :

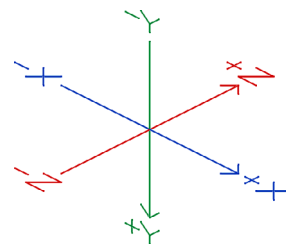
- To speed up parts authoring by providing a library of components which can be incorporated into several parts
- To allow rendering software to make substitutions of curved components

Within this reference material the available primitives are categorised into :

- [Rectilinear primitives](#)
  - [Two-dimensional](#)
  - [Three-dimensional](#)
- [Curved primitives](#)
  - [Two-dimensional](#)
  - [Three-dimensional](#)
- [Special purpose primitives](#)
  - [Technic axle primitives](#)
  - [Technic bush primitives](#)
  - [Technic connector primitives](#)
  - [Stud primitives](#)
  - [Miscellaneous primitives](#)

Each section contains an overview of the characteristics common to all primitives within that category. Primitives are grouped into classes within each category - one class of primitive serving a similar purpose at different sizes or resolutions. For each class of primitive, a brief description of the purpose of the primitive is provided, with notes on its co-ordinate origin, default size and rules for scaling. A list of the available primitives is shown.

An understanding of the orientation of the co-ordinate axes is essential for authoring a part for LDraw. For reference within this page the axes and their direction is shown in this diagram.



### Rectilinear primitives

These rectilinear elements may be scaled in the {x}, {y} and {z} dimensions to make elements of any size. For example

```
1 16 0 0 0 40 0 0 0 1 0 0 0 20 rect.dat
```

would generate a 80LDu x 40LDu rectangle in the {x,z} plane.

Although the default orientation of the rect.dat primitive is in the {x,z} plane the LDraw language allows for this to be transformed

```
1 16 0 0 0 0 1 0 40 0 0 0 0 20 rect.dat
```

would generate a 80LDu x 40LDu rectangle in the  $\{y,z\}$  plane.

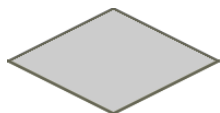
```
1 16 0 0 0 40 0 0 0 0 20 0 1 0 rect.dat
```

would generate a 80LDu x 40LDu rectangle in the  $\{x,y\}$  plane.

## Two dimensional

[Top](#)

rect.dat



### Rectangle with all edges

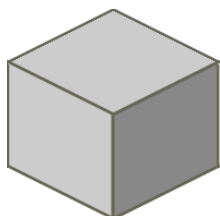
This primitive represents a rectangle in the  $\{x,z\}$  plane and the four edges that bound it. Its origin is at its centre and by default has a size of 2LDu in each of the  $\{x\}$  and  $\{z\}$  dimensions. To avoid matrix arithmetic problems in some renderers, the third dimension ( $\{y\}$  in the default orientation) should be given a scaling factor of 1.

```
1 16 0 0 0 5 0 0 0 1 0 0 0 20 rect.dat
```

## Three dimensional

[Rectilinear primitives](#) [Top](#)

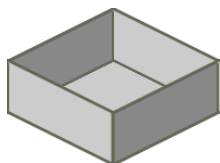
box.dat



### Cuboid with all faces and edges

This primitive is used to define a cuboid. Its origin is the centre of the cuboid and by default has a size of 2LDu in each of the three dimensions.

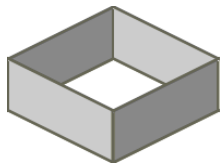
box5.dat



### Cuboid with 5 faces and all edges

This primitive represents a cuboid missing the top  $\{-y\}$  face. Its origin is the centre of the (missing) top face and by default has a size of 2LDu in each of the  $\{x\}$  and  $\{z\}$  dimensions and 1LDu in the  $\{y\}$  dimension.

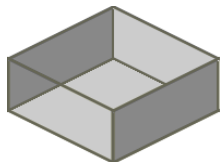
box4.dat



### Cuboid with 4 faces and all edges

This primitive represents a cuboid missing the top  $\{-y\}$  and bottom  $\{+y\}$  faces. Its origin is the centre of the (missing) top face and by default has a size of 2LDu in each of the  $\{x\}$  and  $\{z\}$  dimensions and 1LDu in the  $\{y\}$  dimension.

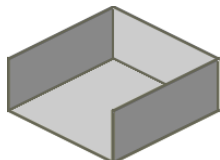
box4t.dat



### Cuboid with 4 adjacent and all edges

This primitive represents a cuboid missing the top  $\{-y\}$  and front  $\{-z\}$  faces, but with all its edges. Its origin is the centre of the (missing) top face and by default has a size of 2LDu in each of the  $\{x\}$  and  $\{z\}$  dimensions and 1LDu in the  $\{y\}$  dimension.

box4-1.dat



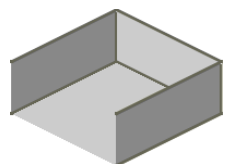
### Cuboid with 4 adjacent faces missing 1 edge

This primitive represents a cuboid missing the top  $\{-y\}$  and front  $\{-z\}$  faces and the edge between those faces. Its origin is the centre of the (missing) top face and by default has a size of 2LDu in each of the  $\{x\}$  and  $\{z\}$  dimensions and 1LDu in the  $\{y\}$  dimension.

box4-4a.dat

### Cuboid with 4 adjacent faces missing 4 edges

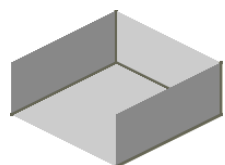
This primitive represents a cuboid missing the top  $\{-y\}$  and front  $\{-z\}$  faces and all the edges of the missing front face. Its origin is the centre of the (missing) top face and by default has a size of 2LDu in each of the  $\{x\}$  and  $\{z\}$  dimensions and 1LDu in the  $\{y\}$  dimension.



box4-7a.dat

### Cuboid with 4 adjacent faces missing 7 edges

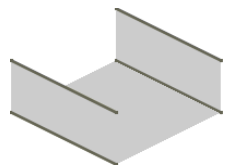
This primitive represents a cuboid missing the top  $\{-y\}$  and front  $\{-z\}$  faces and all the edges of both missing faces. Its origin is the centre of the (missing) top face and by default has a size of 2LDu in each of the  $\{x\}$  and  $\{z\}$  dimensions and 1LDu in the  $\{y\}$  dimension.



box3#8p.dat

### Cuboid with 3 adjacent faces missing 8 edges (in two parallel groups)

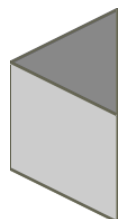
This primitive represents a cuboid missing the top  $\{-y\}$  and left  $\{-x\}$  and right  $\{+x\}$  faces and excludes all the edges of the left  $\{-x\}$  and right  $\{+x\}$  faces. Its origin is the centre of the (missing) top face and by default has a size of 2LDu in each of the  $\{x\}$  and  $\{z\}$  dimensions and 1LDu in the  $\{y\}$  dimension.



tri3.dat

### Right-angled triangular prism with 3 faces

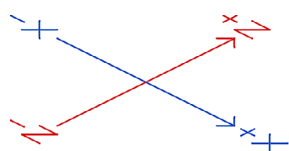
This primitive represents a triangular prism missing the top  $\{-y\}$  and bottom  $\{+y\}$  faces but including all edges. The left  $\{-x\}$  and front  $\{-z\}$  faces are perpendicular. Its origin is the right-angle corner of the (missing) top face and by default has a size of 1LDu in each of the  $\{x\}$  and  $\{z\}$  dimensions and 1LDu in the  $\{y\}$  dimension.



## Curved primitives

LDraw represents curved surfaces as polygons. For circular components two series of primitives are provided.

All the circular primitives are orientated in the  $\{x,z\}$  plane with their origin at the centre of the circle and a default radius of 1 LDu. Primitives are provided for complete circles and for commonly used fractions of a complete circle. Where the naming convention includes a prefix of the form  $n-f$  this indicates the fraction  $(n/f)$  of the circle drawn by the primitive. Where this fraction is less than an entire circle, the primitive starts at  $\{+x,0\}$  and progresses in a counterclockwise direction when viewed from above  $\{-y\}$ .



To avoid matrix arithmetic problems in some renderers, the third dimension ( $\{y\}$  in the default orientation) of two-dimensional primitives should be given a scaling factor of 1.

LDraw circles are normally formed of 16-sided polygons (hexadecagons) - the *regular resolution*. For larger elements, where scaling-up of hexadecagons would give too angular an appearance, a series of *high resolution* primitives based on a 48-sided polygon are available. These may also be used for parts not well suited to a 16-fold symmetry.

These circular elements may be scaled by the same factor in both the  $\{x\}$  and  $\{z\}$  dimensions to make circular elements of greater or less than 1LDu radius. For example

```
1 16 0 0 0 3 0 0 0 1 0 0 0 3 4-4edge.dat
```

would generate a circle in the  $\{x,z\}$  plane with a diameter of 3LDu.

They may also be scaled asymmetrically in the  $x$  and  $z$  dimension to make ellipses.

Although the default orientation is in the  $\{x,z\}$  plane the LDraw language allows for these to be transformed

1 16 0 0 0 0 1 0 3 0 0 0 0 3 4-4edge.dat would generate a circle in the  $\{y,z\}$  plane

1 16 0 0 0 3 0 0 0 0 3 0 1 0 4-4edge.dat would generate a circle in the  $\{x,y\}$  plane

## Two dimensional

[Curved primitives](#) [Top](#)

`n-fedge.dat`

### Circular line segment

This suite of primitives are used for edges which comprise an entire or part circle.



*Currently available primitives :*

Regular resolution (n-f) : 1-8, 1-4, 3-8, 2-4, 3-4, 4-4

High resolution (n-f) : 1-8, 1-4, 2-4, 4-4

`n-fdisc.dat`

### Circular disc sector

This suite of primitives are used for surfaces which comprise an entire or part circle.



*Currently available primitives :*

Regular resolution (n-f) : 1-8, 1-4, 2-4, 3-4, 4-4

High resolution (n-f) : 1-8, 1-4

`n-fndis.dat`

### Inverse of circular disc sector

This suite of primitives pad their matching `n-fdisc.dat` primitives out to the bounding square. They are used to integrate circular elements into rectilinear elements.



*Currently available primitives :*

Regular resolution (n-f) : 1-8, 1-4, 2-4, 3-4, 4-4

High resolution (n-f) : 1-8, 1-4

`n-fringr.dat`

`n-frinrr.dat`

`ringr.dat`

`ringrr.dat`

### Circular ring segment



This suite of primitives are used to generate circular rings or part rings. The numeric suffix  $r$  in the filename indicates the inner radius of the ring - the outer radius is 1LDu greater. For example a `n-fring4` primitive would create a ring with an inner radius of 4LDu and an outer radius of 5LDu.

These primitives are currently undergoing a change of naming convention - names of the form "ringr.dat" are being migrated to "4-4rinrr.dat". Backward compatibility will be maintained.

*Currently available primitives :*

Regular resolution :

r=1 : 1-4ring1, 2-4ring1, ring1

r=2 : 1-4ring2, 2-4ring2, ring2

r=3 : 1-8ring3, 1-4ring3, 2-4ring3, ring3

r=4 : 1-4ring4, 2-4ring4, ring4

r=5 : 1-4ring5, 4-4ring5

r=6 : 1-4ring6, 4-4ring6

r=7 : 1-4ring7, 2-4ring7, ring7

r=8 : 1-4ring8

r=9 : 1-4ring9

r=10 : 1-4rin10, ring10

r=12 : 1-4rin12

r=15 : 1-4rin15

r=16 : 1-4rin16

r=18 : 1-4rin18

r=19 : 1-4rin19

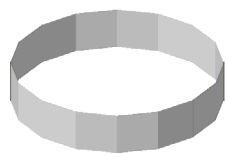
r=38 : 1-4rin38

r=39 : 1-4rin39  
 High resolution :  
 r=3 : 1-4ring3  
 r=4 : 1-4ring4  
 r=9 : 1-4ring9, 4-4ring9

[Curved primitives](#) [Top](#)

### Three dimensional

n-fcyli.dat



#### Circular cylinder

This suite of primitives are used to generate cylinder or part cylinders.

Currently available primitives :

Regular resolution (n-f) : 1-8, 1-4, 3-8, 2-4, 3-4, 4-4

High resolution (n-f) : 1-8, 1-4, 4-4

n-fcyls.dat  
 n-fcyls2.dat

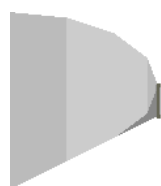


#### Circular cylinder truncated by an angled plane

These primitives are used to generate cylinders or part cylinders which are truncated by a plane which is not perpendicular to the axis of the cylinder. The default angle of the plane is 45 degrees.

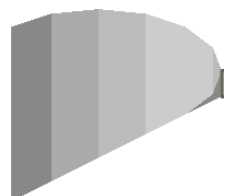
These primitives are notoriously difficult to describe and the user is encouraged to experiment in order to gain a full understanding of their geometry. One technique is to open the primitive in L3Lab, increase the zoom to 12800, deselect BFC and choose random colours. Selecting Show Axes also helps.

1-4cyls.dat



The 2-4cyls.dat primitive is orientated with the perpendicular bounding plane at the top {-y} and the angled bounding plane at the bottom {+y}. In common with the other 2-4xxxx.dat primitives, the {+z} semicircle is represented. By default the truncation plane is y=x-1. The 1-4cyls.dat and 3-8cyls.dat primitives are sub-sections of 2-4cyls.dat

3-8cyls.dat



The 1-4cyls2.dat primitive is orientated with the perpendicular bounding plane at the top {-y} and the angled bounding plane at the bottom {+y}. Unlike the other 1-4xxxx.dat primitives, the {-x,+z} quadrant is represented. The truncation plane is y=x.

The following observations may help :

- 1-4cyls and 1-4cyls2 are complements. If you put them together correctly they make a 1-4cyli.

```
1 15 0 0 0 1 0 0 0 1 0 0 0 1 1-4cyls.dat
```

```
1 14 0 1 0 -1 0 0 0 -1 0 0 0 1 1-4cyls2.dat
```

- 1-4cyls2 is actually hidden in 2-4cyls, which could be constructed like

```
1 16 0 0 0 1 0 0 0 1 0 0 0 1 1-4cyls.dat
```

```
1 16 0 0 0 0 0 -1 0 1 0 1 0 0 1-4cyli.dat
```

```
1 14 0 1 0 1 0 0 0 1 0 0 0 1 1-4cyls2.dat
```

2-4cyls.dat



1-4cyls2.dat

Currently available primitives :

Regular resolution : 1-4cyls, 3-8cyls, 2-4cyls, 1-4cyls2

n-fconr.dat  
 n-fconrr.dat



#### Circular cone

This suite of primitives are used to generate circular cones or part cones. The numeric suffix r in the filename indicates the inner radius of the cone - the outer radius is 1LDu greater. By default the cone is 1LDu high in the {+y} dimension with the origin at the centre of the outer diameter. For example a n-fcone4 primitive would create a cone with an inner radius of 4LDu and an outer radius of 5LDu.

Currently available primitives :

Regular resolution :

- r=0 : 1-4con0, 4-4con0
- r=1 : 1-4con1, 4-4con1
- r=2 : 1-4con2
- r=3 : 1-4con3
- r=4 : 1-4con4
- r=8 : 4-4con8
- r=10 : 1-4con10
- r=12 : 4-4con12

High resolution :

- r=5 : 1-4con5
- r=6 : 1-4con6
- r=9 : 1-4con9

### Circular torus

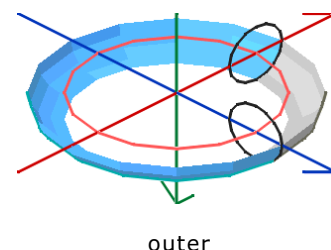
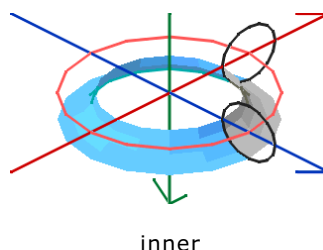
This suite of primitives are used to generate circular torus sections. By default all these primitives produce a torus with a major radius of 1LDu, so typically need to be scaled up in the {x} and {z} dimensions. The second and third characters of the filename *ff* denotes the sweep of torus, as an inverse fraction (01=1/1, 02=1/2, 04=1/4, 08=1/8, 16=1/16, 48=1/48). The fourth character denotes the section of a torus (i=inner, o=outer, q=tube - the combination of 2 inner and 2 outer sections). The last four characters of the file name *rrrr* denote the tube radius in LDu (1333=0.1333, 3333=0.3333).

Currently available primitives :

Regular resolution :

- Inner : t04i0625, t04i1304, t04i1333, t04i3333
- Outer : t04o0625, t04o1304, t04o1333, t04o1429, t04o1538, t04o3333
- Tube : t04q0625

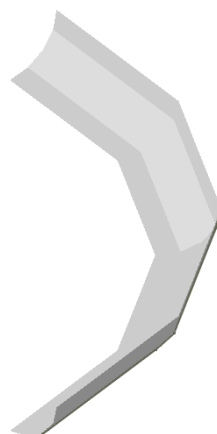
These diagrams (based on an idea of Paul Easter's) illustrate the relationship of the torus primitives to the major (orange) and minor (black) radii.



tffirrrr.dat  
tfforrrr.dat  
tffqrrrr.dat



inner



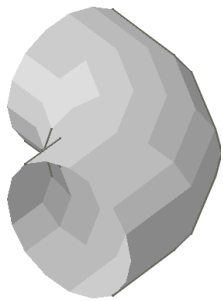
outer



tube

1-4ccyli.dat

### Circular torus (obsolete)



This single primitive pre-dates the creation of the *tffirrrr* and *tfforrrr* suite of primitives and will not be augmented. It produces a quarter torus with a major radius of 2.5LDu and a tube radius of 1LDu.

It should be superseded by *t04q4000.dat*.

*n-fsphe.dat*

### Sphere section



*Currently available primitives :*

Regular resolution : 1-8, 2-8, 4-8, 8-8

*1-8sphc.dat*

### Spherical corner



This primitive represents one octant (eighth) of a sphere, centered at the origin  $\{0,0,0\}$  of radius 1.414, truncated by the sides of a cube with a vertex at  $\{1,1,1\}$ . The boundaries of the resulting surface are circular and fit with *1-4edge.dat*.

## Special purpose primitives

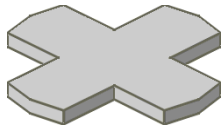
### *Technic axle primitives*

[Top](#)

These primitives represent various components of the technic axle and its matching hole. They are orientated in the  $\{x,z\}$  plane. Except where noted below, and in the  $\{y\}$  dimension only, these primitives should not be scaled.

*axle.dat*

### Technic axle section



This primitive is comprises a 1LDu long section of technic axle, including its ends. It may be scaled in the  $\{y\}$  dimension to produce an axle of any length.

*axlehole.dat*

### Technic axle hole - closed



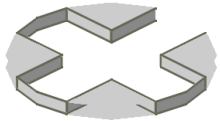
This primitive produces a technic axle hole with the disc ends and all sides. It may be scaled in the  $\{y\}$  dimension as necessary.

*axleho11.dat*

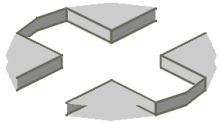
### Technic axle hole - reduced



This primitive produces a technic axle hole with reduced teeth including the disc ends and all edges. It may be scaled in the  $\{y\}$  dimension as necessary.

**axleho14.dat****Technic axle hole - open one side**

This primitive produces a technic axle hole with one side omitted. It may be scaled in the {y} dimension as necessary.

**axleho15.dat****Technic axle hole - open two sides**

This primitive produces a technic axle hole with two opposite sides omitted. It may be scaled in the {y} dimension as necessary.

**axleho16.dat****Technic axle hole tooth**

This primitive represents one tooth of a technic axle hole.

**axleend.dat****Technic axle end**

This primitive is used to produce the "plus-shaped" cross section of a technic axle.

**axleho10.dat****Technic axle hole end**

This primitive used to produce the fill the "plus-shaped" cross section of a technic axle to the bounding circle.

**axleho17.dat****Technic axle sides**

This primitive is comprises a 1LDu long section of outer sides of a technic axle. It may be scaled in the {y} dimension.

**axleho18.dat****Technic axle sides**

This primitive is comprises a 1LDu long section of technic axle, without any ends. It may be scaled in the {y} dimension as necessary.

**axleho12.dat****Technic axle outer edges**

This primitive produces the edges at the ends of the "plus-shaped" technic axle.

**axleho13.dat****Technic axle hole edges**

This primitive produces the outer edges of the teeth of a technic axle hole.

**axleho19.dat****Technic axle inner edges**

This primitive produces the inner edges of the "plus-shaped" technic axle.

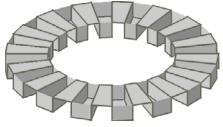
## Technic bush primitives

[Special purpose primitives](#) [Top](#)

These primitives are various representations of the castellated technic bush. They are orientated with the technic axle hole along the  $\{y\}$  axis and should not be scaled.

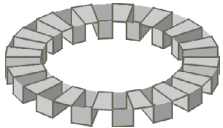
bushlock.dat

**Technic 16-tooth castellation - long teeth**



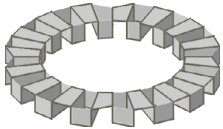
bushloc2.dat

**Technic 16-tooth castellation - regular teeth**



bushloc3.dat

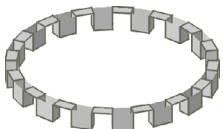
**Technic 16-tooth castellation - regular teeth indented**



This primitive differs from bushloc2.dat in that the teeth are indented in four places to allow it to fit between four adjacent studs.

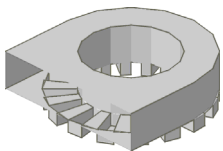
bushloc4.dat

**Technic 16-tooth castellation - short teeth**



steerend.dat

**Curved end to technic 1xn plate with 16-tooth castellation on underside**



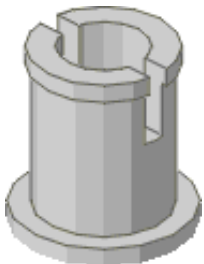
## Technic connector primitives

[Special purpose primitives](#) [Top](#)

These primitives are used to construct technic connector pegs. They are orientated with the technic axle hole along the  $\{y\}$  axis and should not be scaled.

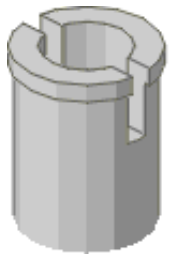
connect.dat

**Technic connector - long with collar**

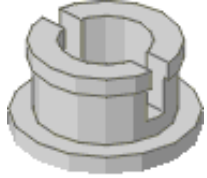


connect2.dat

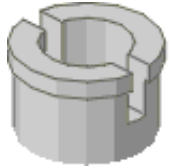
**Technic connector - long without collar**



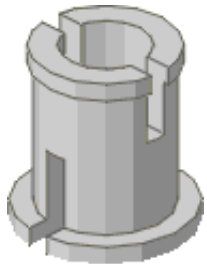
connect3.dat

**Technic connector - short with collar**

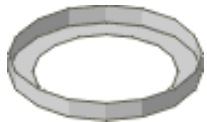
connect4.dat

**Technic connector - short without collar**

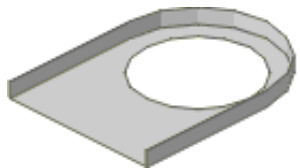
connect5.dat

**Technic connector - long with collar and slot**

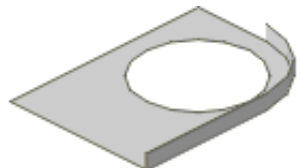
peghole.dat

**Technic connector hole**

peghole2.dat

**Technic connector hole - 180 degrees**

peghole3.dat

**Technic connector hole - 90 degrees****Stud primitives**

[Special purpose primitives](#) [Top](#)

Each stud $x$  primitive described below has a matching low resolution stu $2x$  primitive, used by the fast-draw mode of renderers - these stu $2x$  primitive should never be used in part files. An additional

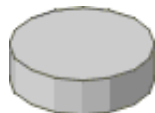
primitive (studline.dat) is used to substitute a single line for studs by the super fast-draw mode of LDraw.

All stud primitives are modelled with the solid disc uppermost, even if designed for the underside of parts (stud3, stud4, stud8). The origin is as the base of the stud and they extend for 4LDU in the  $\{-y\}$  dimension. To use for the underside, the primitive need to be inverted in the  $\{y\}$  dimension, viz.

```
1 16 x y z 1 0 0 0 -1 0 0 0 1 stud3.dat
```

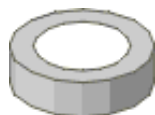
These primitives should not be scaled in the  $\{x\}$  or  $\{z\}$  dimensions. Ideally they should not be scaled in the  $\{y\}$  dimension either, to allow the accurate substitution of chamfered studs by high-quality renderers [but this rule is flouted in the regular brick files, where stud4 is scaled by 5 to generate the underside tube].

**stud.dat**



**Regular stud**

**studp01.dat**



**Regular stud with white dot**

**stude1.dat**



**Regular electric stud**

The electric contact is on the  $\{-x\}$   $\{-z\}$  corner

**stud2.dat**



**Hollow stud**

**stud2a.dat**



**Hollow stud without edge around base**

**stud3.dat**



**Small underside stud**

This is modelled with the solid disc uppermost. To use for the underside of plates, reference like

```
1 16 x y z 1 0 0 0 -1 0 0 0 1 stud3.dat
```

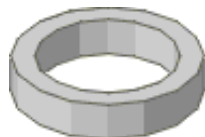
**stud3a.dat**



**Small underside stud without edge around base**

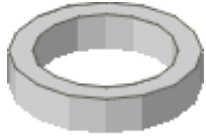
See note for stud3.

**stud4.dat**

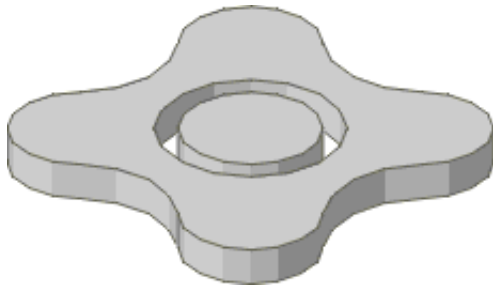
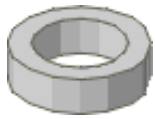


**Ring underside stud**

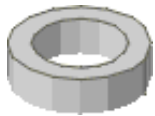
See note for stud3.

**stud4a.dat****Ring underside stud without edge around base**

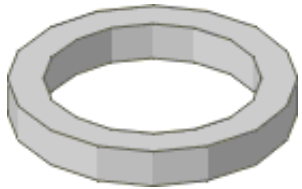
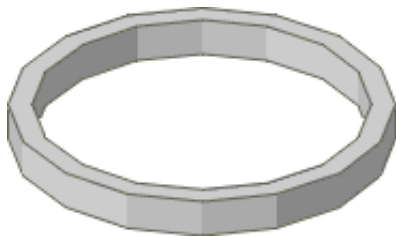
See note for stud3.

**stud5.dat****Scala stud****stud6.dat****Truncated hollow stud**

For use on 2x2 round parts where the stud does not hang over the edge of the part.

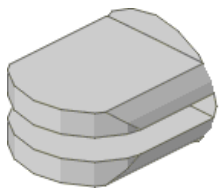
**stud6a.dat****Truncated hollow stud without edge around base**

For use on 2x2 round parts where the stud does not hang over the edge of the part.

**stud7.dat****Duplo hollow top stud****stud8.dat****Duplo hollow underside stud****Miscellaneous primitives**[Special purpose primitives](#) [Top](#)

This section comprises miscellaneous primitives that do not fit neatly into any of the other categories. All are highly specialised and represent components of parts which fit together with each other or other standard parts. As such these primitives are not intended to be re-sized.

**arm1.dat****Cylindrical arm two-fingered hinge**

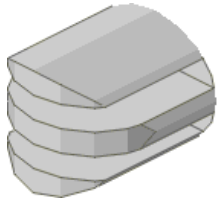


This primitive produces two-fingered hinge component of the cylindrical arm with a radius of 10LDu.

`arm2.dat`

### Cylindrical arm three-fingered hinge

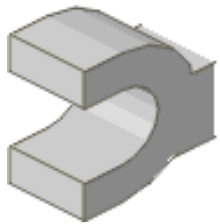
This primitive produces three-fingered hinge component of the cylindrical arm with a radius of 10LDu.



`clip3.dat`

### Horizontal clip

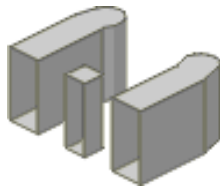
This primitive produces a horizontal clip designed to hold an 8LDu diameter pole or rod.



`clip4.dat`

### Vertical clip - three fingered

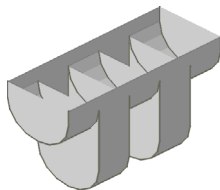
This primitive produces a three-fingered vertical clip designed to hold an 8LDu diameter pole or rod.



`h1.dat`

### Two-fingered plate hinge

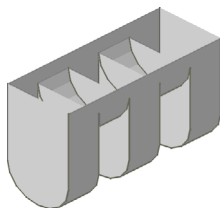
This primitive produces the standard two-fingered hinge with a depth of 8LDu - the thickness of a plate.



`h2.dat`

### Three-fingered plate hinge

This primitive produces the standard three-fingered hinge with a depth of 8LDu - the thickness of a plate.



Last updated 30 November 2002, to include LDraw updates up to 2002-05, by [Chris Dee](#).

[Parts List](#) | [Activity](#) | [Submit](#) | [PT Tools](#) | [PT Reference](#)