

NAME

arch2rad - convert Architrion text file to RADIANCE description

SYNOPSIS

```
arch2rad [ -n ] [ -m mapfile ] [ input ]
```

DESCRIPTION

Arch2rad converts an Architrion text file to a RADIANCE scene description. The material names for the surfaces will be assigned based on the default mapping or the mapping rules file given in the *-m* option. A mapping file contains a list of materials followed by the conditions a surface must satisfy in order to have that material.

For example, if we wanted all surfaces for blocks with RefId "thingy" and Color 152 to use the material "wood", and all other surfaces to use the material "default", we would create the following mapping file:

```
default ;
wood (RefId "thingy") (Color 152) ;
```

All surfaces would satisfy the first set of conditions (which is empty), but only the surfaces in blocks with RefId "thingy" and Color 152 would satisfy the second set of conditions.

Each rule can have up to one condition per qualifier, and different translators use different qualifiers. In *arch2rad*, the valid qualifiers are *Layer*, *Color*, *Face* and *RefId*. A condition is either a single value for a specific attribute, or an integer range of values. (Integer ranges are specified in brackets and separated by a colon, eg. [-15:27], and are always inclusive.) A semicolon is used to indicate the end of a rule, which can extend over several lines if necessary.

The semantics of the rule are such that "and" is the implied conjunction between conditions. Thus, it makes no sense to have more than one condition in a rule for a given qualifier. If the user wants the same material to be used for surfaces that satisfy different conditions, they simply add more rules. For example, if the user also wanted surfaces in blocks with RefId "yohey" with Colors between 50 and 100 to use "wood", they would add the following rule to the end of the example above:

```
wood (Color [50:100]) (RefId "yohey") ;
```

Note that the order of conditions in a rule is irrelevant. However, the order of rules is very important, since the last rule satisfied determines which material a surface is assigned.

By convention, the identifier "void" is used to delete unwanted surfaces. A surface is also deleted if it fails to match any rule. Void is used in a rule as any other material, but it has the effect of excluding all matching surfaces from the translator output. For example, the following mapping would delete all surfaces in the Layer 2 except those with the color "beige", to which it would assign the material "beige_cloth", and all other surfaces would be "tacky":

```
tacky ;
void (Layer 2) ;
beige_cloth (Layer 2) (Color "beige") ;
```

If neither the *-m* nor the *-n* options are not used, *arch2rad* uses the default mapping file `/usr/local/lib/ray/lib/arch.map`. This file simply assigns materials based on color, using the identifiers "c0" through "c255". Appropriate materials for these identifiers are contained in `/usr/local/lib/ray/lib/arch.mat`.

The *-n* option may be used to produce a list of qualifiers from which to construct a mapping for the given Architrion file. If the *-m* option is used also, only those blocks matched in the mapping file will be added to the qualifier list.

DETAILS

Architrion blocks are divided into about 6 polygons. The reference, opposite and end faces must all be quadrilaterals (ie. four-sided polygons), though one or more faces may disappear in certain degenerate cases. The bottom face will usually be a quadrilateral, though it may be written out as two triangles if the face is non-planar or one triangle if there is a degenerate side. The top face is treated the same as the bottom face.

Openings are currently handled using the antimatter material type. An antimatter material called "opening" is defined that "clips" all faces for the current block, and patches the edges of the hole with the material defined for the face "sill". If no rule is given specifically for the sill face, then the most specific material (ie. the material in the latest rule) for this block is used. An antimatter opening will not function properly if there is another surface intersecting it, or rendering is attempted from within the opening. Overlapping openings or openings with shared boundaries will also fail. There is currently no support of Architrion "frame" libraries.

Naming of the output faces is based on layer number, reference id and output block number (sequential from 1 to the total number of output blocks). If there is no reference id name, the layer name is used (if available) instead of the layer number. If there is a reference id number but no name, that is added as well. Names are truncated to the first 12 characters, so the ends of long names may be lost. Also, spaces in names are replaced by underscores ('_'). Finally, the face id is added to the end of the block name for each output polygon. An example identifier for a polygon is:

```
l3.window_overh.3155.ref
```

This would be the reference face of output block number 3155, reference id name "window overhangs" in layer number 3.

EXAMPLE

To create a qualifier list for building.txt:

```
arch2rad -n building.txt > building.qual
```

To translate building.txt into a RADIANCE file using the mapping building.map:

```
arch2rad -m building.map building.txt > building.rad
```

To create an octree directly from an Architrion file using the default mapping and materials:

```
oconv source.rad /usr/local/lib/ray/lib/arch.mat `!\arch2rad building.txt` > building.oct
```

FILES

```
/usr/local/lib/ray/lib/arch.map /usr/local/lib/ray/lib/arch.mat
```

AUTHOR

Greg Ward

SEE ALSO

```
ies2rad(1), oconv(1), thf2rad(1), xform(1)
```