

**NAME**

`rhcopy` - copy ray information into a holodeck

**SYNOPSIS**

```
rhcopy dest_holo [ -u ][ -d ] -h src_holo ..
or
rhcopy dest_holo [ -u ][ -d ] -p src_pic src_zbf ..
```

**DESCRIPTION**

*Rhcopy* adds ray sample data to the existing holodeck *dest\_holo*. In the first form, the ray samples are taken from one or more holodeck files given after the *-h* option. In the second form, the ray samples are taken from one or more RADIANCE picture files and their depth buffers, which must be paired on the command line after the *-p* option.

The *-u* option turns on duplicate ray checking. In some cases, the same ray may already exist in the destination holodeck, and it would be redundant to add it. By default, *rhcopy* does not check for duplicates, because it takes extra time, and in many invocations is not necessary, as when copying into an empty holodeck.

The *-d* option turns off depth checking. Normally, *rhcopy* checks the OBSTRUCTIONS variable of the destination holodeck, and if it is set to True, makes sure that all contributing rays start outside each section. If OBSTRUCTIONS is set to False, then *rhcopy* makes sure that any contributing rays end outside each section. If OBSTRUCTIONS is not set, then this option has no effect. (See the *rholo(1)* man page for a definition of the OBSTRUCTIONS variable.)

*Rcopy* cannot be used to create a holodeck -- use *rholo* for this purpose. For example, to create an empty holodeck, run *rholo* without either the *-n* or *-o* option. Whatever variables are set by *rholo* when the new holodeck is created are the ones that will affect later rendering or viewing. Since the ray sample data may be taken from any source, *rholo* and *rhcopy* may be used together to change certain unalterable holodeck parameters, such as the section grid geometry.

**EXAMPLE**

To take data from an existing holodeck after changing the section grid:

```
rholo new.hdk new.hif
rhcopy new.hdk -h old.hdk
```

To add ray samples from two pictures to the new holodeck:

```
rhcopy new.hdk -p view1.hdr view1.zbf view2.hdr view2.zbf
```

**NOTES**

*Rhcopy* attempts to place the beams in the holodeck in a good order for quick access, but if the data comes from multiple sources, the results may not be optimal. For large holodecks, it is sometimes useful to run the *rhoptimize(1)* program once all runs of *rhcopy* are through.

**AUTHOR**

Greg Ward Larson

**ACKNOWLEDGMENT**

This work was supported by Silicon Graphics, Inc.

**SEE ALSO**

*getinfo(1)*, *pfilt(1)*, *rhinfo(1)*, *rholo(1)*, *rhoptimize(1)*, *rpict(1)*