

## NAME

Tie::RefHash - use references as hash keys

## SYNOPSIS

```
require 5.004;
use Tie::RefHash;
tie HASHVARIABLE, 'Tie::RefHash', LIST;
tie HASHVARIABLE, 'Tie::RefHash::Nestable', LIST;

untie HASHVARIABLE;
```

## DESCRIPTION

This module provides the ability to use references as hash keys if you first `tie` the hash variable to this module. Normally, only the keys of the tied hash itself are preserved as references; to use references as keys in hashes-of-hashes, use `Tie::RefHash::Nestable`, included as part of `Tie::RefHash`.

It is implemented using the standard perl `TIEHASH` interface. Please see the `tie` entry in `perlfunc(1)` and `perltie(1)` for more information.

The `Nestable` version works by looking for hash references being stored and converting them to tied hashes so that they too can have references as keys. This will happen without warning whenever you store a reference to one of your own hashes in the tied hash.

## EXAMPLE

```
use Tie::RefHash;
tie %h, 'Tie::RefHash';
$a = [];
$b = {};
$c = \*main;
$d = \"gunk";
$e = sub { 'foo' };
%h = ($a => 1, $b => 2, $c => 3, $d => 4, $e => 5);
$a->[0] = 'foo';
$b->{foo} = 'bar';
for (keys %h) {
    print ref($_), "\n";
}

tie %h, 'Tie::RefHash::Nestable';
$h{$a}->{$b} = 1;
for (keys %h, keys %{$h{$a}}) {
    print ref($_), "\n";
}
```

## THREAD SUPPORT

*Tie::RefHash* fully supports threading using the `CLONE` method.

## STORABLE SUPPORT

*Storable* hooks are provided for semantically correct serialization and cloning of tied reffashes.

## RELIC SUPPORT

This version of `Tie::RefHash` seems to no longer work with 5.004. This has not been thoroughly investigated. Patches welcome ;-)

**LICENSE**

This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself

**MAINTAINER**

Yuval Kogman <nothingmuch@woobling.org>

**AUTHOR**

Gurusamy Sarathy gsar@activestate.com

'Nestable' by Ed Avis ed@membled.com

**SEE ALSO**

perl(1), perlfunc(1), perltie(1)