

FAQ 4.69 How can I use a reference as a hash key?

Source: <http://coding.derkeiler.com/Archive/Perl/comp.lang.perl.misc/2005-11/msg00941.html>

- *From:* PerlFAQ Server <comdog@xxxxxxxx>
 - *Date:* Tue, 15 Nov 2005 11:03:01 +0000 (UTC)
-

This message is one of several periodic postings to comp.lang.perl.misc intended to make it easier for perl programmers to find answers to common questions. The core of this message represents an excerpt from the documentation provided with Perl.

4.69: How can I use a reference as a hash key?

(contributed by brian d foy)

Hash keys are strings, so you can't really use a reference as the key. When you try to do that, perl turns the reference into its stringified form (for instance, "HASH(0xDEADBEEF)"). From there you can't get back the reference from the stringified form, at least without doing some extra work on your own. Also remember that hash keys must be unique, but two different variables can store the same reference (and those variables can change later).

The Tie::RefHash module, which is distributed with perl, might be what you want. It handles that extra work.

Documents such as this have been called "Answers to Frequently Asked Questions" or FAQ for short. They represent an important part of the Usenet tradition. They serve to reduce the volume of redundant traffic on a news group by providing quality answers to questions that keep coming up.

If you are some how irritated by seeing these postings you are free to ignore them or add the sender to your killfile. If you find errors or other problems with these postings please send corrections or comments to the posting email address or to the maintainers as directed in the perlfaq manual page.

Note that the FAQ text posted by this server may have been modified

FAQ 4.69 How can I use a reference as a hash key?

from that distributed in the stable Perl release. It may have been edited to reflect the additions, changes and corrections provided by respondents, reviewers, and critics to previous postings of these FAQ. Complete text of these FAQ are available on request.

The perlfac manual page contains the following copyright notice.

AUTHOR AND COPYRIGHT

Copyright (c) 1997–2002 Tom Christiansen and Nathan Torkington, and other contributors as noted. All rights reserved.

This posting is provided in the hope that it will be useful but does not represent a commitment or contract of any kind on the part of the contributors, authors or their agents.

.

- *Follow-Ups*