

Re: make -U

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2003-08/0037.html>

From: Ruslan Ermilov (ru_at_FreeBSD.org)

Date: 08/02/03

Date: Sat, 2 Aug 2003 20:13:52 +0300

To: Faried Nawaz <fn@hungry.com>

[Moved to -arch]

> *From the message in freebsd-hackers which first introduced
> this patch:*

>

> - *Date: Tue, 29 Jul 2003 09:09:17 -0700*

> - *From: Faried Nawaz <fn@hungry.com>*

> - *Subject: patch to add make -U*

>

> *While working around a port issue (ports/55013), I discovered
> that make couldn't unset variables using make -U. I've written
> a small patch that adds -U functionality, but I haven't tested
> it extensively.*

>

> *<http://web.nilpotent.org/tmp/make.diff.bz2> (~ 3KB unpacked)*

> *against yesterday's -CURRENT code.*

>

The patch looks sane.

> *A simple Makefile I used to test it:*

>

> *-- cut here --*

> *FOO = bar*

>

> *.ifdef FOO*

> *SAY = y*

> *.else*

> *SAY = n*

> *.endif*

>

> *all:*

> *echo \$(SAY)*

> *-- cut here --*

>

> *Try "make -U FOO".*

>

Re: make -U

> *Personally I think this is a reasonable option to implement.*
> *An undefined variable is not the same as a variable which is*
> *defined to be a null string.*

>

I want that everyone understands the effect of the patch: the -U FOO option causes the FOO variable to always be reported by make(1) as undefined, not only in this makefile, but even in sub-make processes, and affects all makefiles wishing to set it -- it just doesn't let them do this, globally.

So the actual meaning of this option is "prevent variable FOO from being set". (The implementation is somewhat different; it causes make(1) to lie that variable being is, but the net effect is the same.)

Given the above, do we really want this option in our make?

Cheers,

--

Ruslan Ermilov	Sysadmin and DBA,
ru@sunbay.com	Sunbay Software Ltd,
ru@FreeBSD.org	FreeBSD committer

-
- application/pgp-signature attachment: stored