

Re: Per-source CFLAGS

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2003-06/0287.html>

From: Marcel Moolenaar (*marcel_at_xcllnt.net*)

Date: 06/22/03

Date: Sun, 22 Jun 2003 03:02:49 -0700

To: arch@freebsd.org

On Sat, Jun 21, 2003 at 11:45:22PM -0700, David Schultz wrote:

> >

> > *Per file compilation options are in direct conflict with make
> > invocator control, by way of it being a makefile writer knob.
> > Put differently: it's a feature for developers, not builders.
> > We already see the problem with that when we define CFLAGS on
> > the make command line, rather than in the environment. I'm
> > not opposed to per-file options, but it seems to push the
> > need to split make invocator knobs from makefile writer knobs.
> > Until we have such seperation, I request that per-file options
> > be made conditional so that make invocators still have control
> > without being powerless.*

>

> *I expect that this feature would not be used except in very
> special cases, and I would be opposed to gratuitous use of it.*

My expectations are less rose-coloured :-)

> *In fact, most of these cases are so special that the relevant
> file probably won't even work without the extra option.
> For example, Peter mentioned a while ago that vfprintf.c was
> causing an ICE unless -O was turned off.*

A #pragma around the function is normally a good way to avoid file-scoped pessimisation.

> *Since these things are only used selectively, it only makes sense
> to disable them selectively. For instance, if we set it on two
> files to temporarily work around a gcc bug, and on another file
> because it's vendor code that we don't want to see warnings for, a
> big knob that says ``Turn off all the special cases'' wouldn't
> make much sense. However, if what you're looking for is the
> ability to say
> GDTOA_WARNINGS=YES
> in your make.conf, that can certainly be done on a case by case
> basis.*

freebsd-arch: Re: Per-source CFLAGS

It's exactly what I not want. It's control without power. Think what a build invocator has to figure out if he wants an uniform build and there are hundreds of makefiles; all possibly containing specialized compiler options on a per-file basis.

> *Would this satisfy your concerns?*

Not really, but don't worry about it. As I said, I'm not apposed to the feature. There are legit cases where it's useful. If it's causing us problems, we'll just deal with it then and there. It's just that I've already been there, and I'm not in a hurry to go back :-)

--

Marcel Moolenaar

USPA: A-39004

marcel@xcllnt.net

freebsd-arch@freebsd.org mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-arch>

To unsubscribe, send any mail to "freebsd-arch-unsubscribe@freebsd.org"