Subject: Re: Finding if a structure tag name is defined?
Posted by David Fanning on Tue, 14 Oct 2003 14:32:36 GMT
View Forum Message <> Reply to Message

William writes:

> I want to know if a given keyword has been passed down via _extra, so
> I said "if (n_elements(e.c_linestyle) ne 0)" which of course doesn't
> work because if it isn't there you can't say e.linestyle (bring on
> perl...). I can use tag names to get all the tags, then search through
> for a match, but that seems silly and ugly. What is the correct way
> to find if a given structure has a certain tag in it? (And why
> couldn't I find the answer buried within DF's site ;-)?

_EXTRA is often a great convenience (and maybe even
a miracle when you are writing object programs), but
is should NOT be used to collect keywords you care
about (at least not in the way you appear to be using it
here). In fact, I'm adding a rule to Coyote's List of Rules
You Violate Only If You Wish Your Program to Break at the
Most Inopportune Moment:

Rule 3495: If you care about a keyword, be sure
you define it somewhere!

Consider the PLOT command. It has 50-60 different keywords
that can be used with it. But if you are writing some
kind of wrapper to the PLOT command that uses a nifty
new plot symbol, you could care less about 49-59 of those
keywords. But you care desperately about the PSYM keyword.
In that case, you MUST define a PSYM keyword for your
program.

It is not fair to write this program and then decide
later than you really care about the LINESTYLE keyword,
too, and decide to fish it out of the _EXTRA structure.
(And what a nightmare, since you will have to look for
all possible spellings of LINESTYLE!) If you care
about it, you define it. Period. :-)

All those other keywords can be collected easily and
simply with _EXTRA. It is a convenience. (And also,
occasionally, a pain in the neck. Maybe you really
want to use _STRICT_EXTRA instead. Think about it.)

_EXTRA gives you more time to go drink coffee. But it doesn't
take away the responsibility of being a conscientious
programmer. :-)

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155
