Subject: Re: undefined keyword variables
Posted by davidf on Mon, 01 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

J.D. Smith (jdsmith@astro.cornell.edu) writes the kind of
article I wish I could write more often when he says:

> It's really not a difference between built-in and compiled routines,
> just well-written and poorly written routines. Back when I first
> noticed this phenomenon of built-in routines recognizing undefined
> variables, I immediately knew that RSI programmers had access to some
> argument functionality we in compiled-land did not. Thus was
> arg_present() born. I can now write a compiled routine which can:
>
> 1) Discern if a keyword is passed at all.
> 2) Discern if a keyword is passed with a value.
> 3) Discern if a keyword is passed which has scope in the passing level
> (by reference).
>
> Both 2 & 3 can be simultaneously true. So, since the introduction of
> arg_present, we can make programs which handle undefined submitted
> keywords gracefully, in whatever way necessary. This doesn't mean we
> *will*. Here is an example which demonstrates the various
> possibilities. Note that keyword_set is a really a subset of
> n_elements, and so isn't explicity included, though it can be useful.

> I can easily produce a routine which fails on some keywords and not on
> others when passed undefined variables. So can RSI. The problem is
> there isn't always a correct thing to do... maybe an error is actually
> appropriate in some cases, but consistency should be policy.

Good stuff, here.

Cheers,

David

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