Subject: Re: Optional parameters
Posted by David Fanning on Tue, 24 Feb 2004 01:05:20 GMT
View Forum Message <> Reply to Message

Michael Wallace writes:

> The learning of IDL continues.... today's question is about optional
> parameters. Specifically, how do you make a parameter optional? I can
> find stuff in the IDL documentation regarding position parameters and
> keywords, but I haven't found anything yet about optional parameters.
> So, how do you create optional parameters and what are common gotchas
> with using them?

An optional parameter can be either a positional parameter or a keyword parameter, although the general rule of thumb is that positional parameters are always required and keyword parameters are always optional. The rule is broken frequently, especially with positional parameters and less so with keyword parameters (having to suffer the wrath of Coyote if you violate the "keyword is optional" dictum probably accounts for the decreased frequency of the latter).

What makes a parameter "required", oddly enough, does not depend on how it is defined or used (e.g., is it an input or output parameter). Rather, what makes a parameter "required" is how you respond to its absence or presence. It is required if you notice it is not there when you need it and issue an error message. It is optional if you notice it is not there when you need it and you arbitrarily define a default value for it.

Here is a simple program with a required positional parameter named "junk":

```
PRO TEST, junk
  On_Error, 1
  IF N_Elements(junk) EQ 0 THEN Message, 'JUNK is required.'
  Print, junk
END
```

Here is the same program with an optional positional parameter named "junk":

```
PRO TEST, junk
  On_Error, 1
  IF N_Elements(junk) EQ 0 THEN junk = 3
  Print, junk
END
```

Here is the same program with an optional keyword parameter

named "junk":

```
PRO TEST, JUNK=junk
  On_Error, 1
  IF N_Elements(junk) EQ 0 THEN junk = 3
  Print, junk
END
```

I'm not going to show you a program with a required keyword parameter because no one in his right mind (with the possible exception of the good folks at RSI) would write one. :-)

Note that if the parameter has a *binary* quality (that is it is 0 or 1, yes or no, on or off, etc.) then you can check the parameter with KEYWORD_SET. All other parameters should be checked with N_ELEMENTS. This is a subtle point that only bites you when your equipment is on it way to Mars. :-)

Good programming!

Cheers,

David

P.S. Let's just say the only gotchas with optional parameters is forgetting to check whether they are defined or not, and using the wrong function to check whether they are defined or not. Remember, you can *never* go wrong with N_ELEMENTS. :-)

--
David Fanning, Ph.D.
Fanning Software Consulting
Coyote's Guide to IDL Programming: http://www.dfanning.com/
