Subject: Re: IDL versus MATLAB : could you help me ?????
Posted by Mark Hadfield on Mon, 03 Dec 2001 21:56:24 GMT
View Forum Message <> Reply to Message

From: "Nabeel" <nabeel@mathworks.com>
>> Well, in MatLab you must either keep track of the
>> order of input parameters, or you have to parse the
>> input manually (of type which is done in the plotting routines:
>
> Not anymore - there's a relatively common programming pattern in MATLAB
> that makes use of structures to address this functionality. ...

Thanks for that info, Nabeel. I am a long-time IDL user in a discipline where Matlab is the de facto standard so every so often I try Matlab out. Invariably I find myself stuck trying to do things in an IDL-ish way. The issue of optional arguments is a particularly sticky area. When I explain to colleagues what I am trying to do they invariably say, "Huh? Why would you want to do that?" The next time I dabble with Matlab I will try out your suggestion.

But what I would like to know is, does Matlab support anything like IDL's keyword inheritance? It is extremely useful for wrapper functions. I don't know if you know that means so I will try to illustrate with an example. The IDL plot command (like Matlab's plot, i think) accepts a huge list of optional arguments. In IDL they are implemented as keywords, eg you might write

```
   plot, x, y, xrange=[10,20], yrange=[0,3], color=10B
```

Let's say I want a routine that, by default, plots data in red. (Let's also assume that through a deep and mystical knowledge of IDL colour-specification I have associated red with value 10B. Hmmm maybe this isn't such a good example but I will persist.) Then I could write plot_red like this

```
 pro plot_red, p1, p2, _EXTRA=extra
   plot, p1, p2, COLOR=10B, _EXTRA=extra
end
```

When plot_red is called all its keywords are bundled up (into a structure in this case) and passed to plot. If there is no COLOR keyword amongst them then the "COLOR=10B" keyword is passed to plot, but if there is a COLOR keyword then it takes precedence.

This is extremely useful once you get the hang of it. They key thing is that in defining plot_red I don't have to know or care what keywords are accepted by plot, other than COLOR.

> I'm not sure about what other programming languages
> everyone here is familiar with, but if you aren't familiar
> with structures you can think of them as hash tables, with
> the key being specified after the "."

IDL users will find that Matlab structures are a lot like IDL ones, but there is a significant difference. IDL structures are immutable, i.e. once you've made one the only way to add a tag is to destroy the old structure and create a new one. This is what the following code does:

```
s = {param1: 42}
s = create_struct(s, 'param2', "Nabeel'}
```

In Matlab, you can extend an existing structure any time you like, and the syntax is simpler, eg:

```
S.param1 = 42;
S.param2 = 'Nabeel';
```

So Matlab wins on flexibility here. Of course, there may be efficiency advantages to IDL's approach

---
Mark Hadfield
m.hadfield@niwa.cri.nz  http://katipo.niwa.cri.nz/~hadfield
National Institute for Water and Atmospheric Research

--