Subject: Re: rebin question
Posted by Mark Hadfield on Mon, 25 Mar 2002 22:07:38 GMT
View Forum Message <> Reply to Message

"Mark Fardal" <fardal@coral.phys.uvic.ca> wrote in message
news:yj1zo0xta36.fsf@coral.phys.uvic.ca...
> hradilv.nospam@yahoo.com (Vince) writes:
>> Is it possible to define a function or procedure in IDL that can take
>> an arbitrary number of arguments...
>
> Sure. Something like this:
>
> function my_rebin, a, $
> arg0, arg1, arg2, arg3, arg4, arg5, arg6, arg7, arg8, arg9
> ;continue to heart's content
>
> narg = n_params()-1
> command = 'result = round(rebin(float(a)'
> for i = 0, narg-1 do begin
> command = command + string(', arg',i,format='(a,i0)')
> endfor
> command = command + '))'
> junk = execute(command)
> return, result
> end
>
> I imagine there are limits placed on this technique by the maximum
> number of arguments to an IDL procedure, or the maximum length of a
> string EXECUTE can handle.

The maximum number of arguments to an IDL procedure is far higher than any sane programmer would want to use. The maximum length of an EXECUTE'd string used to be a significant limitation in earlier versions of IDL (I think it was 64, later 256) but in 5.4 or 5.5 it was also increased to an effectively infinite value.

When writing wrapper procedures of this sort (which I do quite often, for various reasons) I prefer to use this form

```
pro myfoo, p1, p2, ...
   case n_params() of
      0: foo
      1: foo, p1
      2: foo, p1, p2
      ...
   endcase
end
```

I have gone as far as handling 15 positional parameters, but I don't
think I've ever used more than 4.

--
Mark Hadfield
m.hadfield@niwa.co.nz Ka puwaha et tai nei
http://katipo.niwa.co.nz/~hadfield Hoea tatou
National Institute for Water and Atmospheric Research (NIWA)