

---

Subject: Re: call\_procedure with a dynamically created arguments list?

Posted by **JD Smith** on Mon, 09 Oct 2006 18:37:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

On Fri, 06 Oct 2006 14:54:38 -0700, Dometz wrote:

> Unfortunately, I do have to care about bicyclist with brain damages.  
> Since I am going to wrap IDL functions, I need to still return the same  
> kind of error message as when the IDL function is called directly. I  
> wish I wouldnt have to deal with invalid code but I have to.  
> All I am trying to do is to make the wrapper functions as transparent  
> to the user as possible. I dont care about avoiding errors if the user  
> enters invalid arguments or not the right number of arguments. I just  
> want to provide the user with the same error messages than he would  
> have gotten if he didnt use the wrapper functions.

Languages like lisp allow you to "advise" functions to add this type of functionality. I think Dometz wants a single magic super-duper routine, which can take any number of arguments/keyword parameters, log them, then pass them transparently on to the "real" routine. This is a tall order. You can't concatenate all arguments into an array: what if one is a float, and one is a string? If you really, really, really want to go this route, you might try something like the one below (and a similar wrap\_function), limited to 25 arguments maximum.

JD

```
pro wrap_procedure,routine,v1,v2,v3,v4,v5,v6,v7,v8,v9,v10, $  
    v11,v12,v13,v14,v15,v16,v17,v18,v19,v20, $  
    v21,v22,v23,v24,v25,_REF_EXTRA=e  
on_error,2  
log_all,routine, v1,v2,v3,v4,v5,v6,v7,v8,v9,v10, $  
    v11,v12,v13,v14,v15,v16,v17,v18,v19,v20, $  
    v21,v22,v23,v24,v25,e  
narg=n_params()-1  
case narg of  
  0: call_procedure,routine,_STRICT_EXTRA=e  
  1: call_procedure,routine,v1,_STRICT_EXTRA=e  
  2: call_procedure,routine,v1,v2,_STRICT_EXTRA=e  
  3: call_procedure,routine,v1,v2,v3, $  
      _STRICT_EXTRA=e  
  4: call_procedure,routine,v1,v2,v3,v4, $  
      _STRICT_EXTRA=e  
  5: call_procedure,routine,v1,v2,v3,v4,v5, $  
      _STRICT_EXTRA=e  
  6: call_procedure,routine,v1,v2,v3,v4,v5,v6, $  
      _STRICT_EXTRA=e  
  7: call_procedure,routine,v1,v2,v3,v4,v5,v6,v7, $
```

```

    _STRICT_EXTRA=e
8: call_procedure,routine,v1,v2,v3,v4,v5,v6,v7, $
    v8,_STRICT_EXTRA=e
9: call_procedure,routine,v1,v2,v3,v4,v5,v6,v7, $
    v8,v9,_STRICT_EXTRA=e
10: call_procedure,routine,v1,v2,v3,v4,v5,v6,v7, $
    v8,v9,v10,_STRICT_EXTRA=e
11: call_procedure,routine,v1,v2,v3,v4,v5,v6,v7, $
    v8,v9,v10,v11,_STRICT_EXTRA=e
12: call_procedure,routine,v1,v2,v3,v4,v5,v6,v7, $
    v8,v9,v10,v11,v12,_STRICT_EXTRA=e
13: call_procedure,routine,v1,v2,v3,v4,v5,v6,v7, $
    v8,v9,v10,v11,v12,v13,_STRICT_EXTRA=e
14: call_procedure,routine,v1,v2,v3,v4,v5,v6,v7, $
    v8,v9,v10,v11,v12,v13,v14,_STRICT_EXTRA=e
15: call_procedure,routine,v1,v2,v3,v4,v5,v6,v7, $
    v8,v9,v10,v11,v12,v13,v14,v15,_STRICT_EXTRA=e
16: call_procedure,routine,v1,v2,v3,v4,v5,v6,v7, $
    v8,v9,v10,v11,v12,v13,v14,v15,v16,_STRICT_EXTRA=e
17: call_procedure,routine,v1,v2,v3,v4,v5,v6,v7, $
    v8,v9,v10,v11,v12,v13,v14,v15,v16,v17,_STRICT_EXTRA=e
18: call_procedure,routine,v1,v2,v3,v4,v5,v6,v7, $
    v8,v9,v10,v11,v12,v13,v14,v15,v16,v17,v18, $
    _STRICT_EXTRA=e
19: call_procedure,routine,v1,v2,v3,v4,v5,v6,v7, $
    v8,v9,v10,v11,v12,v13,v14,v15,v16,v17,v18,v19, $
    _STRICT_EXTRA=e
20: call_procedure,routine,v1,v2,v3,v4,v5,v6,v7, $
    v8,v9,v10,v11,v12,v13,v14,v15,v16,v17,v18,v19,v20, $
    _STRICT_EXTRA=e
21: call_procedure,routine,v1,v2,v3,v4,v5,v6,v7, $
    v8,v9,v10,v11,v12,v13,v14,v15,v16,v17,v18,v19,v20,v21, $
    _STRICT_EXTRA=e
22: call_procedure,routine,v1,v2,v3,v4,v5,v6,v7, $
    v8,v9,v10,v11,v12,v13,v14,v15,v16,v17,v18,v19,v20,v21, $
    v22,_STRICT_EXTRA=e
23: call_procedure,routine,v1,v2,v3,v4,v5,v6,v7, $
    v8,v9,v10,v11,v12,v13,v14,v15,v16,v17,v18,v19,v20,v21, $
    v22,v23,_STRICT_EXTRA=e
24: call_procedure,routine,v1,v2,v3,v4,v5,v6,v7, $
    v8,v9,v10,v11,v12,v13,v14,v15,v16,v17,v18,v19,v20,v21, $
    v22,v23,v24,_STRICT_EXTRA=e
25: call_procedure,routine,v1,v2,v3,v4,v5,v6,v7, $
    v8,v9,v10,v11,v12,v13,v14,v15,v16,v17,v18,v19,v20,v21, $
    v22,v23,v24,v25,_STRICT_EXTRA=e
endcase
end

```

---