
Subject: Re: Pass struct with pointer to array to/from dlm
Posted by [Nigel Wade](#) on Tue, 18 May 2004 08:31:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ed Wright wrote:

```
> To: IDL
> From: Ed Wright
>
> I have a requirement to pass a structure between IDL and a DLM. The
> structure has the form:
>
> struct CELL
>
> { SpiceCellDataType dtype; /* An enum */
>   SpiceInt      length; /* long */
>   SpiceInt      size; /* long */
>   SpiceInt      card; /* long */
>   SpiceBoolean  isSet; /* long */
>   SpiceBoolean  adjust; /* long */
>   SpiceBoolean  init;
>   void          * base;
>   void          * data; };
>
> My problem concerns the fields 'base' and 'data'. Normally, these
> contain pointers to arrays of either ints, doubles or an array of
> strings. Is it possible to pass from a DLM to IDL (and back) a struct
> where one field points to an array?
>
> As always,
> Ed Wright
>
```

Yes, it's perfectly possible (but don't do anything with it in IDL). I do quite a lot of it.

What I do is store it in a byte array (char in C), of length sizeof(pointer) to take care of 32/64 bit portability. My code is probably overkill safetywise, but I like to be sure that what is being passed back into C is really a C pointer.

This code copies the pointer PI_handle to an IDL byte array.

```
/*
 * create an IDL byte array to hold the pointer.
 * this is returned in the third positional parameter.
 * the callback for destruction of the variable will free the
 * allocated memory.
 */
```

```

    UCHAR *store;
    PI_Strategy *pointer;

    dims[0] = sizeof(pointer);
    store = (UCHAR *)malloc(dims[0]);
    memcpy(store, &pointer, dims[0]);
    new_array = IDL_ImportArray(1, dims, IDL_TYP_BYTE,
        (UCHAR *)store, idl_PI_strategy_cb, NULL);
    IDL_VarCopy(new_array, parameters[3]);

```

To extract it from argv[0]:

```

    PI_Strategy *pointer;

    IDL_ENSURE_ARRAY(argv[0]);
    IDL_EXCLUDE_EXPR(argv[0]);
    if ( argv[0]->type != IDL_TYP_BYTE ||
        argv[0]->value.arr->n_dim != 1 ||
        argv[0]->value.arr->dim[0] != sizeof(PI_Strategy *) ||
        argv[0]->value.arr->free_cb != idl_PI_strategy_cb )
        IDL_Message(IDL_M_NAMED_GENERIC, IDL_MSG_LONGJMP, "arg 1 is not a
valid PI handle.");

    /*
    * copy the embedded pointer from the IDL byte array into
    * the C pointer.
    */
    memcpy(&pointer, argv[0]->value.arr->data, sizeof(pointer));

```

```

void idl_PI_strategy_cb(unsigned char *arg) {

```

```

    free(arg);

```

```

}

```

```

--

```

Nigel Wade, System Administrator, Space Plasma Physics Group,
 University of Leicester, Leicester, LE1 7RH, UK
 E-mail : nmw@ion.le.ac.uk
 Phone : +44 (0)116 2523548, Fax : +44 (0)116 2523555
