
Subject: Re: Returning a struct containing variable-length arrays

Posted by [franzpx125](#) on Tue, 16 Sep 2008 21:11:43 GMT

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

On 16 Set, 15:37, Mike <Michael.Mill...@gmail.com> wrote:

> On Sep 15, 6:28 pm, franzpx125 <franzpx...@gmail.com> wrote:

>

>> Hi!

>

>> Within my DLM, I need to return a struct containing two arrays but the

>> length of the arrays is known only at run-time. Which is the correct

>> way to write the DLM code?

>

> You can put a variable length array into a structure by putting a

> pointer to the array in the structure:

>

> result = { x: x, y: y, ptr: ptr_new() }

> ...

> result.ptr = ptr_new(data)

> return, result

>

> I don't know if there are any DLM issues that will affect this, but it

> works well for variable length arrays in objects and other structures.

>

Thanks for your reply. I don't know if it's the correct way to perform this task, anyway I've tried the following workaround and it works.

I'll post the code if you're interested in:

```
IDL_VPTR    idl_out_struct;
```

```
unsigned int* rev_x;
```

```
double*      rev_y;
```

```
unsigned int  num_el;
```

```
void*        s;
```

```
IDL_LONG     tmp_dims[IDL_MAX_ARRAY_DIM];
```

```
IDL_MEMINT   offset;
```

```
double*      y_ptr;
```

```
unsigned int* x_ptr;
```

```
char*        s_data;
```

```
int          i;
```

```
/* .... omitted code .... */
```

```
// Format struct for IDL output:
```

```
REV_tags_dims[0] = 1;
```

```
REV_tags_dims[1] = num_el;
```

```

s = IDL_MakeStruct("REV", REV_tags);

tmp_dims[0] = 1;

s_data = (char *)IDL_MakeTempStruct(s, 1, tmp_dims,
&idl_out_struct, 0);

// Get the field of the structure:
offset = IDL_StructTagInfoByName(s, "Y", IDL_MSG_LONGJMP, NULL);
// Get a pointer to that location:
y_ptr = (double *)(s_data + offset);
// Store values into array:
for ( i = 0; i < num_el; i++)
*(y_ptr++) = rev_y[i];

// Get the field of the structure:
offset = IDL_StructTagInfoByName(s, "X", IDL_MSG_LONGJMP, NULL);
// Get a pointer to that location:
x_ptr = (unsigned int*)(s_data + offset);
// Store values into array:
for ( i = 0; i < num_el; i++)
*(x_ptr++) = rev_x[i];

// Return output in IDL Format:
return idl_out_struct;

```

Any criticism is well accepted.

Regards,
Brun Francesco
