Subject: Re: IDL_MakeStruct()
Posted by Nigel Wade on Wed, 14 Jul 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Chutter wrote:
>
> I have nested structures in some "c" code which I wish to access in IDL
> via Call_External.

I have only ever done this using LINKIMAGE and DLMs. I can't comment on whether
it works with CALL_EXTERNAL. I create a temporary variable in and return the
structures in it.

> Supposedly, the type field of the structure can be:
>
> void *type; /* This may be either a pointer to another
> structure definition, or a simple IDL
> type code (IDL_TYP_*) cast to void
> (e.g. (void *) IDL_TYP_BYTE). If this
> field is NULL, it indicates that IDL
> should search for a structure of the
> given name and fill in the pointer to
> its structure definition. */
>
> from external/export.h
>

This means that there must be a named structure which already exists in IDL,
then IDL will put in the pointer to its definition. I've only done it where
the C code creates the named structure, I don't know if it will look up
a named structure created within IDL, I expect it would.

> What is the syntax for using a pointer to another structure definition?
> The following simple example uses NULL for the type, but I want to a
> pointer to another structure definition so that I can choose the value
> for the name field.

From my experience I think this is what happens.

If you leave type as NULL, then when you create the structure IDL will search for a named structure with the same name as the tag name. If you set type to be the result of a call to IDL_MakeStruct then it will use that structure definition and call it by the name in tag name. So you can use a different name for the tag.

For example, suppose I define the tags for the structures as:

```
 static IDL_STRUCT_TAG_DEF substruct_tags[] = {
  {"P", 0, (void *) IDL_TYP_INT},
  {"Q", 0, (void *) IDL_TYP_INT},
  {0}
 };

 static IDL_STRUCT_TAG_DEF s_tags[] = {
  { "MATRIX", matrix_dims, (void *) IDL_TYP_FLOAT},
  { "SUBSTRUCT", substruct_dims, NULL},  /* NULL is a placeholder */
  {0}
 };
```

The following will work, and will create FOO with tags MATRIX and SUBSTRUCT, where SUBSTRUCT contains P and Q:

```
 s = IDL_MakeStruct("SUBSTRUCT", substruct_tags);
 s = IDL_MakeStruct("FOO", s_tags);
```

This will fail because when FOO is built there is no structure called SUBSTRUCT

```
 s = IDL_MakeStruct("BAR", substruct_tags);
 s = IDL_MakeStruct("FOO", s_tags);
```

However, this will work:

```
 s = IDL_MakeStruct("BAR", substruct_tags);
 s_tags[1].type = s;
 s = IDL_MakeStruct("FOO", s_tags);
```

FOO still contains tags MATRIX and SUBSTRUCT even though SUBSTRUCT was built from the structure named BAR.

I've only ever done this inside a LINKIMAGE/DLM routine, so I don't know what happens when you call IDL_ImportNamedArray from CALL_EXTERNAL.

>
> Thanks for help.
>

No problem.

--

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 2523568, Fax : +44 (0)116 2523555