Subject: Re: callable IDL and structures
Posted by Sebastian Moeller on Tue, 09 Apr 2002 08:00:48 GMT
View Forum Message <> Reply to Message

Hello Mr. Fanning

this really is a nice and impressivly quick news group. Thank you very much for answering.

David Fanning wrote:
[schnipp]
Yes, this is correct (on both counts). Ten minutes of looking through
on-line help doesn't come up with anything, but I know
it is documented *somewhere*. This is an alternative way to
access the fields of a structure. But if they are available
this way, they are sure as heck available via the fields names
as well. I'd like to see the code you are using to "access" them
by their names.
[schnapp]
OK, here we go... This is the part in the C++ project. It is supposed to define the named structure str

```
typedef
struct
{
   int testarr[50];
   int field2;
} teststr;

IDL_MEMINT d1[10];
d1[0]=1;
d1[1]=50;
IDL_STRUCT_TAG_DEF sd[4];
sd[0].name="testarr";
sd[0].type=(void*)IDL_TYP_LONG;
sd[0].dims=(IDL_MEMINT*)d1;
sd[0].flags=NULL;
sd[1].name="field2";
sd[1].type=(void*)IDL_TYP_LONG;
sd[1].dims=NULL;
sd[1].flags=NULL;
sd[2].name=NULL;
sd[2].type=NULL;
sd[2].dims=NULL;
sd[2].flags=NULL;


IDL_MEMINT d[3];
d[0]=1;
```

```
        for(int k=0; k<50; k++)
        {
           teststr.testarr[k]=k;
        }
        teststr.field2=10;
        void* ss=IDL_MakeStruct("str",sd);



var=IDL_ImportNamedArray("teststr",1,d,IDL_TYP_STRUCT,(UCHAR*)(&teststr),0,ss);

        int err=IDL_ExecuteStr("help, /structure, teststr");

/* this line gives the whole information for the structure, including
the tag names and tag types as defined in this c fragment (teststarr and
field2). Also the data contained in the structure (the array containing
the subscript values and 10 for field2) is shown correctly.*/

        err=IDL_ExecuteStr("print, teststr.field2");

/*this line gives an error message about field2 not beeing defined in
teststr, even thoug the preceedind command showed that teststr contained
the tags testarr and field2*/

        err=IDL_ExecuteStr("print, teststr.(where(TAG_NAMES(teststr) EQ
'testarr'))");

/*this line actually works, it prints the content of the array (numbers
from 0 to 49). It is a rude attempt to bring the structure tag names
back into the game, and a pretty obfusceted way to write "print,
teststr.(0)". A few defines along the lines of: #define testarr (0),
would save the day...*/
```

To sum it up we are able to pass the data to IDL. And IDL has all the meta-information about the structure's contents (tags, types and data). The tag names can even be seen from the TAG_NAMES function, but attempts to address the tags directly fail.

I hope this information serves to illustrate the problem more deeply.

Ahoi
Sebastian Moeller

P.S.: Thios is my second attempt to post the reply, as the first seems to be lost somewhere...