
Subject: EDG: Using IDL_MakeTempStruct() vs. IDL_ImportArray()

Posted by [Peter Rodriguez](#) on Tue, 10 Feb 2015 22:11:51 GMT

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

Hi,

My objective is to create within C an IDL_TYP_STRUCT to be returned to IDL.

The EDG and Kling's book cover IDL_ImportArray(), but that costs me coding both for the C & IDL_STRUCT_TAG_DEF structures, so that data may be mapped.

I've been successful just using IDL_MakeTempStruct(strIDLdef, 1, &one, &vIDL, IDL_TRUE) having defined my strIDLdef and populating the structure manually via,

```
char *pIDL=vIDL->value.s.arr->data +
IDL_StructTagInfoByName(strIDLdef,(char*)sTAG,IDL_MSG_LONGJMP,&vTAG);
Then employing CASE(vTAG->type) to stride and store within the data space appropriately.
```

However, upon return to IDL the structure fields are only accessible by index not tagname.

Although the in-built tag_names() works!

```
IDL> o=my_make_struct("blablabla")
```

```
IDL> help,o
```

```
** Structure <743af8>, 7 tags, length=832, data length=830, refs=1:
```

```
sSCAN      STRING   'PPI'
iANT_HGT_m  INT      320
nSWEEPs    LONG      3
fRAYres_deg FLOAT    0.500000
fBINres_km  FLOAT    0.250000
fSWEEP_angle_deg_arr
            FLOAT    Array[40]
sSWEEP_ISO8601_arr
            STRING   Array[40]
```

```
IDL> help,o.sSCAN
```

```
% Tag name sSCAN is undefined for structure <Anonymous>.
```

```
% Execution halted at: $MAIN$
```

```
IDL> help,o.(0)
```

```
<Expression>  STRING   = 'PPI'
```

```
IDL> print,tag_names(o)
```

```
sSCAN iANT_HGT_m nSWEEPs fRAYres_deg fBINres_km fSWEEP_angle_deg_arr
sSWEEP_ISO8601_arr
```

So as a work-around, I've written an IDL wrapper to rebuild/copy the structure based on determining the SIZE(/TYPE) of each field.

Does anyone have insight, how in C may I manually set each structure tagname string? The magic that IDL_ImportArray() apparently does... Something along the lines of tweaking vIDL->value.s.sdef I surmise because from the EDG "the implementation of structure definitions is not public information".

Thanks, Peter Rodriguez
