
Subject: Returning C struct to IDL

Posted by [K Banerjee](#) on Thu, 09 May 2002 21:54:39 GMT

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

Folks,

Here's the platform info:

IDL version 5.3

RedHat Linux

g++ 2.95.2 and g++ 2.96

I have to get IDL to use a C function to read a data file. This C function returns a structure to IDL. I am using IDL's linkimage facility.

Four of the fields of the structure are arrays of type IDL_STRING whose lengths depend on the data file to be read, so these lengths are not known until run time.

Using g++ 2.95.2, this is how I had things set up:

```
static IDL_LONG dims_user_header[IDL_MAX_ARRAY_DIM];
dims_user_header[0] = 1;
dims_user_header[1] = userHeaderArrayLength;
```

(userHeaderArrayLength has been determined previously from the data file.)

```
IDL_STRUCT_TAG_DEF headerTags[] =
{
  {"VERS", 0, (void *) IDL_TYP_STRING},
  {"DATATYPE", 0, (void *) IDL_TYP_INT},
  {"T", 0, (void *) IDL_TYP_LONG},
  {"X", 0, (void *) IDL_TYP_LONG},
  {"Y", 0, (void *) IDL_TYP_LONG},
  {"Z", 0, (void *) IDL_TYP_LONG},
  {"VOXDIMS", dims_4, (void *) IDL_TYP_LONG},
  {"USERHEADER", dims_user_header, (void *) IDL_TYP_STRING},
  // The first of the 4 variable length arrays --^
  .
  .
  .
  {0}
};
```

Next I define a C struct to match the headerTags array:

```
typedef struct
{
    IDL_STRING vers;
    short dataType;
    IDL_LONG t;
    IDL_LONG x;
    IDL_LONG y;
    IDL_LONG z;
    IDL_LONG voxDims[4];
    IDL_STRING userHeader[userHeaderArrayLength];
    .
    .
    .
} vbHeader;
```

I populate all the fields of the C struct and then have the lines:

```
void *psDef = IDL_MakeStruct(NULL, headerTags);
IDL_LONG ilDims[IDL_MAX_ARRAY_DIM];
ilDims[0] = 1;
IDL_VPTR ivReturn = IDL_ImportArray(1, ilDims, IDL_TYP_STRUCT,
(CHAR *) theHeaderActual, NULL, psDef);
return ivReturn;
```

(The implementation of my C function to read the data file is based on "Calling C From IDL" by Mr. Ronn Kling.)

Everything works as I expect. The problem arises when I try to compile this function with g++ 2.96. g++ 2.96 has a problem with the line:

```
IDL_STRING userHeader[userHeaderArrayLength];
```

The specific error message is:

size of member `userHeader' is not constant

So I decided to dynamically allocate memory for the userHeader[] array and changed the troublesome struct field to:

```
IDL_STRING *userHeader;
```

Later on I have:

```
theHeader->userHeader = new IDL_STRING[userHeaderArrayLength];
```

where theHeader is a pointer to the C struct. I then go ahead and populate the struct (I am confident I am populating the

C struct properly), go through the same steps to return the struct to IDL. However, garbage ends up in the userHeader array of the IDL structure. It seems to me that I can not properly "marry" a C struct to an IDL structure when the C struct has pointer fields. I have not come across any example of where a C struct with pointer fields are returned to IDL.

I have perused the IDL External Development Guide, but have not found a solution.

Any ideas as to how I can solve my problem? Is it possible to return a C struct with pointer fields to IDL?

Thanks.

K. Banerjee
