

---

Subject: Passing Structures with Pointers with Call\_External  
Posted by [MajorSetback](#) on Tue, 10 Aug 2004 20:06:42 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

I am using IDL Version 6.0 (linux x86 m32) on Red Hat Linux release 9.

I declare an array of structures thus.

```
function MakeFloatPlane,numrows,numcolumns
    temp={Rows:long(numrows),Columns:long(numcolumns),Data:fltarr r(numrows,numcolumns)}
    return,temp
end
```

```
Planes=replicate(MakeFloatPlane(nY1,nX1),sNumberOfPlanes)
```

I then pass the array to C code through Call\_External thus.

```
Result=Call_External('SharedLibrary.so','CFunction_cw',sNumberOfPlanes,Planes,/unload)
```

The C code is as follows.

```
typedef struct FloatPlane_Struct
```

```
{
    long   Rows;
    long   Columns;
    float  **Data;
} FloatPlane;
```

```
extern "C" long CFunction_cw(int argc, void *argv[])
```

```
{
    long   INumberOfPlanes;
    FloatPlane *fppPlanes;
    float  *fpData;

    INumberOfPlanes=((long *) (argv[0]));
    fppPlanes=((FloatPlane *) (argv[1]));
    fpData=((float *) (argv[3]));

    fprintf(stderr, "INumberOfPlanes=%d\n", INumberOfPlanes);
    fprintf(stderr, "fppPlanes->Rows=%d\n", fppPlanes->Rows);
    fprintf(stderr, "fppPlanes->Columns=%d\n",
fppPlanes->Columns);
    fprintf(stderr, "fppPlanes->Data=%d\n", fppPlanes->Data);
    fprintf(stderr, "fpData=%d\n", fpData);
    fprintf(stderr, "**fpData=%f\n", *fpData);

    return 1;
}
```

I get the expected values for INumberOfPlanes, fppPlanes->Rows, fppPlanes->Columns, fpData and \*fpData. However, I cannot interpret

the result I get for fppPlanes->Data.

If I add

```
fprintf(stderr, "fppPlanes->Data[0]=%d\n", fppPlanes->Data[0]);
```

idlde crashes, presumably due to a memory write error in the C code.  
Is there any way to stop idlde crashing under such circumstances?

My main question is this. Is there a way to retrieve the IDL variable  
Planes[i].Data within CFunction\_cw?

---