Subject: Problems with IDL strings
Posted by jicicuendez on Wed, 08 Oct 2003 10:39:49 GMT
View Forum Message <> Reply to Message

I am in trouble when trying to connect idl with C. I have a c function that produces certain strings and I want to copy them back into an idl array of strings. The c functions produces eleven 28-character long strings and when I use call_external I provide and idl string 11-element array with 28 character long each. What happens is there must be something strange with the memory I provide to the function because after calling the function I get the same string copied everywhere in the IDL array. Testing the contents within the c function the same thing happens.

I provide my commented code to see if anyone has a clue or bumped into this problem before. I wish that RSI would give more info about this things. Documentation is pretty bad!!

Many thanks,
Juan

```
long convert_to_time_ticks(int argc, void *argv[])
{

 long status, cfi_utc[4];
 double *decimal_time, decimal[2];
 long n_data;
 char dut1e[9];
 char utc_string[28];
 int i;
 IDL_STRING *myString;

 decimal_time = (double *)argv[0];
 n_data = *((long *)argv[1]);
 myString = (IDL_STRING *)argv[2];
 for(i=0; i<n_data; i++){

  decimal[0]=decimal_time[i];
  decimal[1]=0.0;

  // function to transform time in decimal days to UTC string
  status=ml_pmjd(cfi_utc, decimal, utc_string, dut1e);

            // copying each string to an array of IDL_string
structure
  strcpy(myString[i].s, utc_string); //Seems OK

            // Checking copied contents and addresses
```

```
  printf("in structure: %s\n",myString[i].s); //Seems OK
            // The memory addresses change-->Seem to be ok
  printf("memory address%d \n", &(myString[i].s));

       }
       // Doing the loop again in once
       for(i=0; i<n_data; i++){

            // Here is the problem, when I print out the contents
of
            // the structure again all the values are the same and
            // equal to the last UTC time read from the function
  printf("in estructure: %s\n",myString[i].s);
  printf("direccion %d \n", &(myString[i].s));

 }

       return status;

}
```