Subject: Passing strings to CALL_EXTERNAL
Posted by Mark Rivers on Fri, 12 Oct 2001 12:50:09 GMT

View Forum Message <> Reply to Message

Folks,

I have learned that the definition of an IDL_STRING structure (see "export.h") on 32 bit machines is going to change in IDL 5.5 so that the string length will be "int" rather than "unsigned short". This means that any existing shareable objects or DLLs that are called from IDL will not work with IDL 5.5 if you are passing strings to them, they will need to be rebuilt.

For my work I decided it was not a good idea to have multiple versions of DLLs and .so files, so I have changed my IDL wrapper routines so that they never pass strings to external code, but convert all strings to byte arrays before the CALL_EXTERNAL (and back to strings afterwards for outputs).

Just a "head's up" for those interested.

Here is the relevant section of export.h from IDL 5.4. In IDL 5.5 IDL_STRING_SLEN_T will become int.

```
/*
 * On most 32-bit machines, sizeof(IDL_STRING) is 8 bytes because we
 * get perfect packing (no holes) in the structure. This comes at the
 * cost of strings limited to 64K in length.
 */
typedef unsigned short IDL_STRING_SLEN_T;
#define IDL_STRING_MAX_SLEN 65534
#endif

typedef struct {  /* Define string descriptor */
  IDL_STRING_SLEN_T slen; /* Length of string, 0 for null */
  short stype;   /* type of string, static or dynamic */
  char *s;   /* Addr of string */
} IDL_STRING;
```

Mark Rivers