
Subject: Keeping you informed

Posted by [Karlo Janos](#) on Mon, 21 Jul 2008 11:39:46 GMT

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

rtowler@gmail.com wrote:

- > I'm guessing here, but your method probably returns a 2 element
- > variant array where each element is a variant containing doubles (in
- > other words a variant of variants). IDL doesn't handle variants
- > well. It can handle a variant that contains a single type, but it
- > cannot handle variants containing mixed types or as in your case
- > variants containing variants.

I contacted the support of XtremeFringe and here is the answer. So I think you are guessing right...

When you use the XFLib COM interface, with only one output (PUMD) the MyXtremeLibObject returns a SAFEARRAY that stores the double matrix. IDL (like MATLAB) marshals automatically for you this SAFEARRAY as a DBLARR(128, 128) (in C++ you should use the SafeArrayGetElement() function). The problem arises when you want to use a function that returns two arguments. In this case the MyXtremeLibObject returns a VARIANT containing two SAFEARRAY (like an array of SAFEARRAY). The MATLAB COM client supports this through the use of cells, if you have a look into the DemFT.m file, you will see that the XFLib COM server is returning a cell array of two elements. In C++ you must use SafeArrayGetElement() two times, first to get access to the first SAFEARRAY and then use again SafeArrayGetElement() to get access to the second SAFEARRAY.

Summarizing the use of XFLib from IDL as a COM server:

- 1) You must check if IDL has an equivalent to the MATLAB "cell" type. In this case it is possible that IDL will make the marshalling from VARIANT of SAFEARRAY to this type, like in MATLAB (have a look in DemFT.m)
- 2) Other thing you can try is to use a DBLARR(128, 128, 2) type (a pointer to an array of DBLARR), and check if IDL makes the marshalling correctly.

Cheers

Karlo
