

Re: How to get an OLE array of objects in Perl?

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.ole/2006-08/msg00168.html>

- *From:* "Alexander Nickolov" <agnickolov@xxxxxxxx>
 - *Date:* Mon, 21 Aug 2006 10:03:24 -0700
-

While I have some rudimentary knowledge in Perl, I don't think it's sufficient for your purposes... Anyway, from the OLE side you probably want a VARIANT with the following type: VT_BYREF | VT_ARRAY | VT_DISPATCH. This is a reference to a safe array of IDispatch pointers. How you construct the safe array in Perl is beyond me, however...

Note this is an untypical argument to pass to such a method. Normally a C++ client would pass a pointer to an uninitialized VARIANT for the method to fill in. Therefore this might not work at all. FWIW the method signature itself is unfriendly for non-C++ developers...

--

=====
Alexander Nickolov
Microsoft MVP [VC], MCSD
email: agnickolov@xxxxxxxx
MVP VC FAQ: <http://www.mvps.org/vcfaq>
=====

<kingskippus@xxxxxxxx> wrote in message
news:1155931922.368652.16910@xx

I'm really scratching my head over this. I posted it over in a Perl group hoping someone familiar with OLE could help, and now I'm posting it here hoping someone here might be familiar with Perl and can help.

I have an OLE object with a method that takes an [out] parameter (a parameter that gets changed within the OLE object method) that is an array of other OLE objects. What I'd like to do is to get that array of other OLE objects and iterate through them, printing out a property from each one. I don't know that much about OLE objects, can someone please help? I've been Googling for hours and feel like I'm close, but I can't find anything for that last little bit.

Here's what I need using semi-pseudocode that I know won't work right:

Re: How to get an OLE array of objects in Perl?

```
use Win32::OLE;
my $tree = Win32::OLE->new( 'Tree.Component' );
my @nodes;

# GetNodes is a method of TreeComponent that is defined like this:
# int GetNodes([out] TreeNode[])
# where TreeNode is another OLE object type

$tree->GetNodes(\@nodes); # This doesn't work...
for $node (@nodes) {
print $node->{Name}."\n"; # Name is a property of TreeNode objects

}
```

No error or warning is generated, but the @nodes array doesn't have any elements in it. However, I do know it has elements, because I've tested this function in another utility, and it returns 89 TreeNode objects.

Here's a piece of code that I found on another site that the author claims to work, but not quite for what I need. This code gets a double back:

```
use Win32::OLE;
use Win32::OLE::Variant;
my $arg4 = Variant( VT_R8 | VT_BYREF, 0.0);
my $service = Win32::OLE->new( '3rdPartyComponent.Service' );
$service->foo( 31, 4000, 28000, $arg4 );
# That last parameter is a double, whose value gets altered within
# and passed back out of the foo method of the $service object.
```

...But I don't want doubles, I want an array of OLE objects, and not being very familiar with OLE, I'm totally stumped. I *think* that the answer is that I have to pass some kind of Variant to the function. That's what this guy does to get a double value passed out of the OLE method that takes an [out] parameter. But if I do that, what kind of Variant do I pass in? And how do I get access to the OLE objects I get back?

I sure would appreciate any help anyone can give me on this!