
Subject: re:DDE on IDL

Posted by [harmer](#) on Thu, 26 May 1994 11:58:17 GMT

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

Thanks

to everyone who helped me out regarding my enquiry about DDE on IDL (viz Mark Hadfield). Below is a posting of the information Mark (Hadfield) gave me:

In article <harmer.3.2DDCACB8@ph.und.ac.za> harmer@ph.und.ac.za (MARK HARMER) writes:

> I am attempting to write a C application which interacts with IDL
> using the IDL DDE routines (viz dde_getservers(), dde_gettopics() ...)
> under windows.
> ...
> The problem I have is that 'dde_getservers()' produces a null result
> when DDESRVR.EXE is running. In fact, with the exception of Word for
> Windows, every DDE server application that I have tried (which is
> quite a few) is not recognised as being 'out there' by a call to
> 'dde_getservers()'.
> This seems to indicate that these applications are not using the DDE
> conventions expected of them by IDL, or I am doing something wrong.

DDE_GETSERVERS() has never worked for me either. I've just assumed it was a bug. (I never noticed that it worked with Word for Windows. I just tried it now and it returned a string array with a single element: "#352:(0001)". I wonder what that's supposed to mean.)

Anyway, the problem with DDE_GETSERVERS()--and I'm sure that's where the problem lies, not with your or anybody else's applications--isn't fatal. You just have to make sure that the DDE server in question is running by some other means. The other IDL DDE routines work, although some don't respond to error conditions in the way the manual says they should.

I have used Visual Basic to write a simple DDE server that supplements IDL's MS-Windows capabilities. With this I can launch a Windows application (without going through DOS) and copy IDL strings to the Clipboard. The Visual Basic programming was quite straightforward. I have also used DDE from IDL to control multiple file conversions in Excel. So it can certainly be done.

You're welcome to e-mail me for more info.

=====

Mark Hadfield

hadfield@greta.niwa.cri.nz

NIWA Marine (Taihoru Nukurangi)

NIWA.GRETA:HADFIELD

Wellington, New Zealand

From: Mark Hadfield <HADFIELD@GRETA.NIWA.CRI.NZ>
To: HARMER@ph.und.ac.za
Date: Wed, 25 May 1994 11:04:00 +1300
Subject: Gidday Mark
X-Gateway: iGate, (WP Office) vers 4.04 - 1046
Message-Id: <2DE289D6.67D4.0001@GRETA.NIWA.CRI.NZ>

Gidday Mark

>> You just have to make sure that the DDE server in
>> question is running by some other means.
>
> what exactly do you mean 'by some other means'? Whenever I played
> with DDE on IDL I had the DDE server in question running already.

I didn't have anything elegant in mind when I wrote 'some other means'. I guess the possibilities are:

- 1) Rely on the user to start the DDE server before running the IDL procedure (and/or to deal with the error message you get when you try to converse with a non-existent server).
 - 2) Write a macro in a suitable programming language or batch language that starts the DDE server whenever you start IDL.
 - 3) Attempt a DDE conversation with the server in question and, if this fails, trap the error, start the server and try again. (I haven't tried this, but it SHOULD be possible.)
- > As I understand DDEs, one runs 'dde_getservers()' which returns an
> array of server names (one for each of the available DDE servers)
> which could be the funny string that Word returned?

No. The service name for Word for Windows is "WinWord", the service name for Excel is "Excel", etc.

> These server
> names have to be quoted in any further DDE transactions (
> dde_gettopics(SERVER), dde_getitems(SERVER),
> dde_request(SERVER,topic,item)) with the DDE server. So
> 'dde_getservers()' not returning anything is quite a problem, one
> needs the server name to identify the server for transactions.

Sure. But you would need to know it anyway to pick the right one out of the DDE_GETSERVERS() list. Depending what you want to do with DDE, you also need to know quite a lot about the functionality of the server. For example,

there is no standard syntax for commands accepted by DDE_EXECUTE, although there are some loose conventions (eg. DDE_EXECUTE commands accepted by Excel and WinWord are basically commands in the application's macro language).

I think DDE_GETSERVERS() is just intended to be a method for establishing whether the server you want is running.

> Your Visual Basic DDE server starts up some Windows applications?

I call the DDE server application "Helper". I'll describe it a bit further below. It can accept an IDL string and execute it as a Windows command via the Visual Basic Shell function (similar to Windows API WinExec).

> How do you do that? Is the Clipboard regarded as a particular DDE server?

No. Helper can also accept an IDL string and send it to the Visual Basic Clipboard object (which is an interface to the Windows Clipboard).

>> I have also used DDE from IDL to control multiple file conversions
>> in Excel.

>
> Same question as above?

Several IDL procedures are included below. Some comments:

XLSTOCSV Procedure is straightforward. Fails with an error if Excel not running. The procedure deletes any file with the same name as the output (CSV) file to avoid a "File exists. Do you want to replace ..." prompt. This is a general problem with DDE--user prompts that occur while executing a command need to be cleared manually.

SPAWNWIN IDL procedures are very simple. The DDE server application TOCLIP helper.exe (service name Helper) maintains two topics Launcher and Clipper. A string sent to Launcher via DDE_EXECUTE is executed; a string sent to Clipper via DDE_EXECUTE is saved to the Clipboard. This is a somewhat unusual approach. Following the conventions of Excel and WinWord, one would have a single topic called System that might accept command strings of the form:

```
[LAUNCH("An IDL string")]  
[CLIP("Another IDL string")]
```

But I couldn't be bothered writing the Visual Basic code to parse the commands.

I would be happy to send executables and/or source code for Helper. However Visual Basic code for a DDE server is rather different from the C code that would be required for the same function, because Visual Basic provides a DDE interface that hides the details of interaction with the DDEML library.

The best way to learn how DDE works is to play around with a DDE server application like Excel or WinWord. (Even better, try the Windows text editor PFE--it actually has documentation!) You can use IDL to do this. I also have a couple of applications lying around that provide a graphical interface to DDE commands for testing DDE servers.

Cheers

(LOTS of source code - IDL pros - deleted to save on net traffic, interested parties can email me or the other Mark)

From: Mark Hadfield <HADFIELD@GRETA.NIWA.CRI.NZ>
To: HARMER@ph.und.ac.za
Date: Wed, 25 May 1994 13:49:00 +1300
Subject: Mark
X-Gateway: iGate, (WP Office) vers 4.04 - 1046
Message-Id: <2DE2B0E0.67D4.0000@GRETA.NIWA.CRI.NZ>

Mark

In my previous message I mentioned the text editor PFE as an example of a good DDE server application. As well as being a very good text editor and a well-written Windows application, PFE is also bundled with a DDE tester. You can get it by anonymous ftp from directory ftp/pub/pc/win3/programr at ftp.cica.indiana.edu (129.79.26.27). File name is pfe0506.zip.

Mark

```
=====
Mark Hadfield                hadfield@greta.niwa.cri.nz
NIWA Marine (Taihoro Nukurangi)  NIWA.GRETA:HADFIELD
Wellington, New Zealand
```

In conclusion it is possible to get the IDL DDE functions to communicate with a Visual Basic program (or C program, I have just managed to get a simple example to work) as long as you ignore the fact that dde_getservers() doesn't work.

Mark Harmer
