
Subject: DDE on IDL

Posted by [harmer](#) on Fri, 20 May 1994 12:27:36 GMT

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

Hi

I am attempting to write a C application which interacts with IDL using the IDL DDE routines (viz `dde_getservers()`, `dde_gettopics()` ...) under windows.

I started by using a simple DDE server application written by Borland (1992), in Borland C, which communicates with a DDE client using the 3.1 api DDEML calls.

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. I have been in contact with CreaSo for the last two weeks and their only response was to advise me to get a book by Peter Norton on DDE's (no title, no publisher).

The ideal would be if someone could point me to source code for a skeleton of a DDE server application (compilable under Borland C, preferably) which can communicate with IDL using the IDL DDE routines. Otherwise any help or pointers would be gratefully received.

Mark Harmer

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.
> ...

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)

