

Search: ☒ The Web ☐ Angelfire[Report Abuse](#) « [Previous](#) | [Top](#)
[100](#) | [Next](#) »

GO GET IT!

share: [del.icio.us](#) | [digg](#) | [reddit](#) | [furl](#) | [facebook](#)

Ads by Google

[Excel reporting for R12](#)Do Oracle Finance reporting. Fast, Easy, Drill down. Free Trial.
[www.excel4apps.com](#)[Document X](#)Documentation made easy for .NET assemblies, COM, VB, VBA and DBs
[www.innovasys.com](#)[dog training commands](#)Free shipping on dog training collars. Expert advice. Learn more!
[e-collars.com](#)[Excel Server 2008](#)DIY your ERP, CRM, and MIS using Excel and SQL Server. It's easy!
[www.QinZhe.com/english/index.htm](#)

Site Search

 ☒ This Site ☐ The Web[Get a free Search Engine](#)

Home

Dynamic Data Exchange Interface Details

DDE in Visual Basic, Web Browsers, the Windows Shell, Excel and other applications

Software

[VB DDE Client Control](#)[VB DDE Server control](#)[DataSend Interprocess Control](#)[WebSpy URL Monitor](#)[COMclean Registry Cleaner](#)[File Association Editor](#)[Process Master](#)[VB Recent Files Editor](#)[Dynamic Data Studio](#)[Autorun Registry Editor](#)[HTML and XML Tag Checker](#)[Algol 60](#)

Resources

[Customer support](#)[Application DDE Interfaces](#)[DDE FAQ Information](#)[VB Runtime and Controls](#)[VB and Windows Programming](#)[Shareware Links](#)[Search VB](#)

Stay in Touch

[Home Page](#)[About Us](#)[Feedback](#)[Send E-mail](#)[Contact information](#)

Before you leave ---

Click the links to the left to investigate our software. Most of it is free!

• **Visual Basic**

- [DDE and COM in Visual Basic](#)
- [DDE and Visual Basic.NET](#)

• **The Windows Shell**

- [How the Windows Shell uses DDE](#)
- [Cannot find the file.... \(or one of its components\)...](#)

• **Web Browsers**

- [Web browser DDE](#)

• **Microsoft Excel and Word**

- [Controlling Excel with DDE](#)
- [Controlling Microsoft Word with DDE](#)
- [Why does Excel crash when I link a cell to a DDE server?](#)

• **Other applications**

- [Obtaining data from Reuter's IDNDDE](#)

• **On the main DDE FAQ page**

- [Overviews and detailed articles about DDE and how it used](#)
- [Download links for DDE tools, components and samples](#)
- [Where to find DDE information on MSDN](#)
- [Newsletter signup, tell you friends about this site](#)

• **Related pages**

- [Support for our DDServer and DDClient ActiveX controls](#)
- [File Association Editor](#) The Registry keys use by the Windows Shell for DDE

DDE and COM in Visual Basic

DDE in Visual Basic

[FastCounter by bCentral](#)

The native implementation of DDE in Visual Basic is poor in terms of capability, speed and bugs. DDE is not supported in VB.NET. (Also Known As VB.NOT because it is completely different from VB6.)

When VB is acting either as server or client:

- The built-in VB DDE mechanism is much slower than when the Windows API is used.
- Except for pictures, only string data can be sent and received.
- You cannot use VB native DDE in a Visual Basic authored ActiveX Control.

In addition, on the client side:

- Event driven programming is not possible, all calls are synchronous with a timeout. The VB program cannot continue processing whilst waiting for a reply from the server, it must wait until the server responds or the timeout expires. (Data sent on hot and warm links is not sent in response to an individual request.)
- You cannot turn Notification or a hot link on or off without closing the DDE conversation and opening it again. Every change of LinkMode closes the conversation and starts a new one, which is a time-consuming process. Of course, setting mode 0 simply closes the conversation.
- The LinkOpen event is raised for a destination control (DDE client) every time it attempts to open a DDE conversation. There is no indication of whether the attempt succeeded. The event is raised on startup even if the LinkMode is initially 0.
- If the program is running on Windows 95 it may crash if too many links are made. For details see [Q180624](#) Invalid Page Fault Occurs with More than 34 DDE Channels.
- It misses data on hot or warm links if the server is too fast because it always uses the fAckReq flag (see [The missing data and unreliability problems.](#))

In addition, on the server side:

- The Service and Item names are limited to valid VB names, although DDE itself has no restrictions.
- The Service name is the original name of the executable. The Item name is the name of the control supplying the data. Both are fixed at compile time and cannot be changed.
- Each source form can have a maximum of 128 data items.
- DDE must be enabled when the server starts, by setting the LinkMode property to 1 at design time. Once the program is running DDE can be closed down by setting the property to 0, and re-enabled by setting it to 1 again.
- If another application does a "Wildconnect" your VB server will probably crash. A crash is less likely if links have been previously opened with the server.

The solution to all these problems is to use our [DDClient](#) and [DDServer](#)

controls. You can work asynchronously using Events, it allows the server to send data as fast as it wants and there are no built in limitations. However, even using DDClient messages will be missed if a MsgBox dialog is on display, because it causes all events to be ignored. The answer is to use the MessageBox API call instead. Details of how to do this can be found on the [DDClient support page](#).

You can download DDClient and DDServer from the [the DDE download area](#). Sample native DDE server and client projects are also available from the download area.

COM in Visual Basic

Visual Basic is good for programming clients and servers when the client creates an instance of a server object or application. It has poor support for programming a client or server when the server must run continuously.

On the client side, GetObject() only works in limited circumstances. It cannot work at all with applications such as Internet Explorer which do not register themselves in the Running Object Table (ROT). We have not been able to make it work properly without a file name. Even when it succeeds with a file name it connects to a random instance of the server, when there is more than one. A typical continuously running server is monitoring hardware or other applications and does not use file names in the required way.

An alternative mechanism is to set a reference to the ActiveX EXE server. The Global MultiUse classes can be accessed by the client. Windows will start the server if necessary. The disadvantage of this method is that the client must be recompiled every time the server is changed. Supplying a drop-in server replacement does not seem to be possible.

On the server side, a continuously running server cannot be programmed at all except by an horrendous hack. It is necessary to construct a vtable interface as an array of Longs and write the QueryInterface, AddRef, Release and other interface code in longhand Visual Basic. The Windows API is used to register the interface in the Running Object Table and remove it when the server closes.

The method was described by Matthew Curland in the August 1997 issue of the Visual Basic Programmer's Journal, page 109. There is a much more extensive treatment of the topic in chapter 10 of his book Advanced Visual Basic 6, ISBN 0-201-70712-8. The book includes the code, which is also on the companion CDROM.

[Application DDE interface topics](#) [General DDE FAQ topics](#) [DDE Software downloads](#) [Home page](#)

DDE and Visual Basic.NET

VB.NET has no native support for DDE. You will have to use the API, or alternatively components such as our [DDClient](#) and [DDServer](#) controls.

COM controls do not work properly in the Beta 1 release of VB.NET. Events do not reach the application. Vtable calls are not supported, although this does not affect DDServer and DDClient. We are assured on the .NET newsgroups that these problems will be fixed in the Beta 2 release.

[Application DDE interface topics](#) [General DDE FAQ topics](#) [DDE Software downloads](#) [Home page](#)

How the Windows Shell uses DDE

The Windows Shell uses DDE and the registry to implement the mechanism whereby files with a particular extension are associated with a group of Action Verbs. The Shell uses the information in the registry to construct a command line for the associated application, and optionally to specify a DDE Execute command to send to it. The application is launched with the command line only if the attempt to send a DDE command fails. The full details of the rules and registry entries are on the [File Association Editor](#) page.

If more than one file is to be opened with a running instance of an application, the Shell starts a new DDE conversation for each one. All the conversations are left open until the final Execute command has been sent, at which point all the conversations are closed. This is a rather time-consuming way of doing things, but it does enable the application to determine whether all the file names have been sent. The application must count the number of conversations still open, and process the list of files when it falls to zero.

Supplying a command line is a simple task for which COM is obviously far too cumbersome. But the Shell does more than that with DDE. You can create, modify and delete program groups and their Shortcuts with a DDE interface which is almost unchanged from 16-bit Windows. The link to the specification on MSDN is at [How to get further information](#).

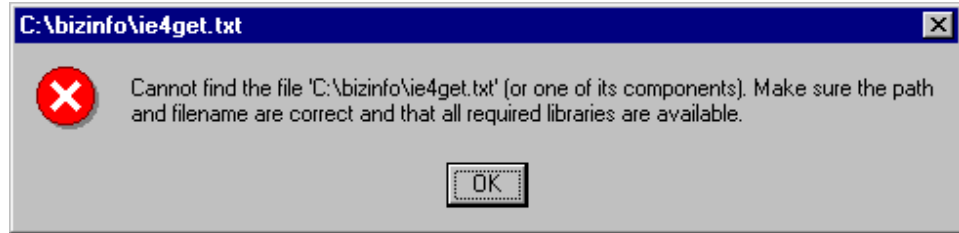
The 32 bit Shell has a COM interface called IShellLink for maintaining Icons and Program Groups. You can use this as an alternative to DDE, but less and simpler code is required using DDE. Even Microsoft has to admit this, if you search MSDN you will find an article entitled "IShellLink-Shortcuts the Long Way" (it concerns shortcuts in the context of Visual Basic). However, 32-bit Shortcuts can be used for purposes other than launching files, for example they can refer to Control Panel items and other types of Shell object. DDE can only be used to manipulate the top level program groups, not groups within groups. If you want to do any of these things you must use IShellLink.

A difference in the DDE interface between 16 and 32-bit Windows is detailed in Knowledgebase article [Q99456](#) Win32 Shell Dynamic Data Exchange (DDE) Interface.

[Application DDE interface topics](#) [General DDE FAQ topics](#) [DDE Software](#)

[downloads](#) [Home page](#)

Cannot find the file.... (or one of its components)...



One possible reason for this message box is that the Windows shell has not been able to establish a DDE conversation according to the specification in the Registry. If this is so, you can edit the file association for the offending verb using the View...Options... dialog of Windows Explorer, File Types tab. It possible that the Application or Topic name is wrong, but more likely you need to uncheck "Use DDE". Alternatively, you can use the [File Association Editor](#), a free download from this web site.

A likely cause of the error, if DDE related, is that another application has modified the entry for files with an extension such as .txt. Notepad does not understand DDE.

[Application DDE interface topics](#) [General DDE FAQ topics](#) [DDE Software downloads](#) [Home page](#)

Web browser DDE

As we have pointed out in the article [When is DDE more suitable?](#), DDE is the ideal way of monitoring the activity of another application. Both Internet Explorer and Netscape Navigator implement a DDE interface for this purpose. You can download our freeware [WebSpy](#) Visual Basic program which uses it.

Netscape Navigator 4

Implements the [Spyglass specification](#). Scroll down for a section on Netscape6.

Internet Explorer

Implements a subset of the [Spyglass specification](#).

One of the most commonly asked questions is how to open a document in a new window instead of reusing an old one. One way of doing it is to start a new instance of IE. Two factors affect whether a window in an existing

instance is reused:

- The setting of the "Reuse windows for launching shortcuts" option under the Tools..Options..Advanced tab of IE. The name is misleading, it affects the issue however the document launch is initiated. If this option is not checked, a new window is opened regardless of the DDE command.
- The DDE command used to open the document. The default one for the http protocol is
"%1",,-1,0,,,,
The third parameter -1 tells IE to reuse the current window. If this is replaced by a zero
"%1",,0,0,,,,
a new window is opened regardless of the "Reuse windows for launching shortcuts" setting.

The fourth parameter 0 does not have any effect, according to [Q160957](#). In the file association for HTML types the DDE command is by default

"file://%1",,-1,,,,

Again, if -1 is replaced by 0 the file is opened in a new window.

The differences between the Internet Explorer DDE interface and the Spyglass specification are detailed in the following Microsoft Knowledgebase articles:

[Q160957](#) DDE Support in Internet Explorer Versions

[Q191508](#) Confusing Return Values from DDE topic WWW_ListWindows

[Q160976](#) Controlling the Currently Running Instance of IE3 via DDE

[Q167826](#) DDE with IE Does Not Work As Expected

There are other differences, not mentioned in any Microsoft article we are aware of. The following table summarizes them.

	Internet Explorer	Netscape Navigator 4
Multiple Browser instances	Each instance of Internet Explorer registers itself as a DDE server. To distinguish between them a client using DDEML can connect with the instance specific service name handle. If this is not done, it is pot luck which	When the first instance of Navigator starts it registers three services, NETSCAPE, nsshell and NSCMDLINE. When a second instance starts it establishes a conversation with the first instance on the service NSCMDLINE and topic WWW_ProcessCMDLINE, the execute command -BROWSER is sent, then the second instance unregisters its services. All DDE communication with Netscape by external

	instance will establish a conversation with a client. If you open a file by double clicking it you cannot control which instance of IE will open it. Instance specific handles are transient and cannot be held in the Registry.	applications is with the first instance started.
Wildconnect	Internet Explorer does not respond to wildconnect. The client must specify both the service and topic names.	Wildconnect is accepted. A conversation on all the supported "WWW_xxx" topics can be established with one DDEML call.
URLs reported by WWW_RegisterURLecho	Internet Explorer reports only the URL requested by the user.	Netscape Navigator reports all URLs loaded including images. All refreshes are also reported, including those in Meta tags and aminated GIFs.
Quotes in strings	Article Q167826 points out that according to the Spyglass specification, quoted strings should be enclosed with the escape sequence "\", not just a double quote mark. Internet Explorer 5 will accept either.	Netscape Navigator 4.7 only accepts the quote mark by itself, the response of other Netscape versions is not known to us.

Netscape 6 Beta releases

The Netscape 6 Beta DDE interface is non-standard and reduced to a single

feature, opening a web page. Connect to service Netscape6 and Topic WWW_OpenURL then issue an Execute command, a request no longer works. The command string must be of the form **"Full path to netscape6"** **"The URL required"** including the quotes. It always opens the new URL in a new window.

Netscape 6 non-Beta

The Netscape 6 non-Beta DDE interface seems to be reduced to absolutely nothing. It is possible to connect to service Netscape6 and Topic WWW_OpenURL, but Execute commands do nothing (although they do not fail), and requests fail.

Opera 5

Opera commits a stupid and unforgivable sin. In addition to responding to the service name "Opera", it also responds to "IEExplore" and "Netscape". What this means is that a program trying to connect to Netscape or Internet Explorer is likely to be connected to Opera instead. Opening a URL is the only DDE operation that seems to work. Opera does not even implement the reduced functionality of Internet Explorer relative to the [Spyglass specification](#). It accepts a conversation on the topic WWW_RegisterURLEcho, but then refuses the Poke required to specify the service name for notifications.

[Application DDE interface topics](#) [General DDE FAQ topics](#) [DDE Software downloads](#) [Home page](#)

Controlling Excel with DDE

Excel is controlled by using it as a DDE server. Excel is a client only when a cell or range has a formula which uses DDE to get the cell contents. DDE allows you to handle multiple instances of Excel, something which is not possible using automation. Each instance offers the "System" topic. The other topics are of the form "[Workbook]Sheet", one for each open workbook regardless of the Excel instance hosting it. The workbook name is the filename of the opened file, if there is one. The filename without square brackets is also available as a topic.

All DDE Service, Topic and Item names are insensitive to case. The DDE Execute command strings are also case insensitive.

If you request "SysItems" on the System topic, you get a list of items available on that topic:

- SysItems - returns the item list.
- Topics - returns "System" and the list of [Workbook]Sheet topics.
- Status - returns "Ready" or "Busy".
- Formats - returns a tab separated list including XTable, cvs, rich text format, bitmap, and text. You can request and poke data in any of

these formats.

- Selection - returns the current selection of the selected worksheet in the format "[Book]Sheet!R1C1:R2C2".
- Protocols - returns "StdFileEditing" and "Embedding".
- EditEnvltems - returns a list starting with "StdHostNames".

The format CF_TEXT must be used for all requests on the System topic.

To manipulate single cells or ranges, use DDE Request and Poke commands on the [Workbook]Sheet topics. The DDE item is the cell name (e.g. r1c1) or range (e.g. r1c1:r10c2).

Single cell data is straightforward. To read the data in a cell, issue a request for the data with the item name being the cell address, such as r1c1. You can also start an advise loop for the cell, known as a hot link. Excel sends the cell contents to the client whenever it changes. To modify the data in a cell, issue a poke transaction with the item name being the cell address as in a request.

There are several ways to populate or read a range, when the target is an array of cells.

- In text form, separate the columns with Tab characters and put each row on a new line, to make a grid like the Excel cells. Poke using CF_TEXT. If you request a range in text format the data is laid out in the way described.
- Separate the columns with commas instead, to make a CSV format (comma separated variables). Register the Clipboard format "CSV" and use that instead of CF_TEXT.
- You can also use XLTable for transferring data when the item name is a range.

You cannot poke or request any part of the active selection if it is being edited in Excel.

Built-in macros can be run on any topic with a DDE Execute command. The command string is the macro enclosed in square brackets, for example [Select("r1c1:r2c2")]. Commands can be concatenated, for example [ALERT("Hello world")][ALERT("Hello again")].

To get help on the macros, download macrofun.exe or macro97.exe from the Microsoft web site from these Knowledgebase articles [Q143466](#) for macro97.exe and [Q128185](#) for macrofun.exe.

To run macros you have written, use the RUN macro like this: [RUN("MyMacro")]. Your macros cannot have parameters. You can write macros to set variables, then execute other macros which examine the variables you have assigned.

In the [DDE Software download area](#) there are sample programs (with full source) which manipulate Microsoft Excel using DDE. The compiled C program is only 17kb, and it does not rely on any external DLLs (apart from the Common Controls which is for the User Interface, not DDE).

[Application DDE interface topics](#) [General DDE FAQ topics](#) [DDE Software downloads](#) [Home page](#)

Controlling Microsoft Word with DDE

The Word DDE interface is similar to that of [Excel](#), with a System topic and a topic for each open file. You cannot connect to the topic of a new document until it has been saved to a file, then it has a topic which is the file name. The commands to send in [] brackets are Word Basic commands. These are the macro commands of Word 7 (Word for Windows 95). If you run Word 7 you can discover the built-in macro commands by recording one of your own.

In Word 97 Word Basic is replaced by VBA and the Word Object Model. The macros you record use object model commands which cannot be run using DDE. Execute commands sent by DDE must still be the old Word Basic ones. The implementation is that a temporary macro called TmpDDE is created using the old macros as elements of the WordBasic object. For example, "CharLeft 1" becomes "WordBasic.CharLeft 1"

Until recently Microsoft had a Word Basic help file installation program on the web site, but it seems to have gone. You can download the file from the Angelfire server only: [click to download wrdbasic.exe](#).

Some of the commands, with no explanation, are in the help file installer Position.exe, which can be downloaded from the Microsoft web site by going to Knowledgebase article [Q112733](#).

[Application DDE interface topics](#) [General DDE FAQ topics](#) [DDE Software downloads](#) [Home page](#)

Why does Excel crash when I link a cell to a DDE server?

When you put a DDE formula in a cell, Excel asks the server for a number of data formats, starting with XLTable. It crashes if the server does not respond in a timely manner, presumably because it is not prepared for the delayed response. The server must therefore immediately reject any data format it cannot supply. Excel eventually asks for CF_TEXT, the least complex clipboard format it uses.

Excel sets up a hot link (advise loop) with the DDE server. It uses the fAckReq flag to stop more data being sent until the last has been acknowledged, which makes sense for a spreadsheet. There is information about fAckReq in the topic [The missing data and unreliability problems](#).

[Application DDE interface topics](#) [General DDE FAQ topics](#) [DDE Software downloads](#) [Home page](#)

Obtaining data from Reuter's IDNDDE

The topic [DDE and COM in Visual Basic](#) explains why Visual Basic can loose updates from Reuter's IDNDDE. Some improvement can be obtained by instructing IDNDDE to ignore the request to wait for acknowledgement of data receipt and to send new data regardless. In the file IDNDDE.INI there is a section for each application which uses it. If you add the line NOREQACK=1 to the relevant section IDNDDE does not wait for acknowledgements from that application.

Deliberately disobeying the correct DDE message protocol is not a good idea. The best solution is to use our [DDClient control](#), which you can download from the [the DDE download area](#). DDClient does it properly, it does not ask the server to wait for acknowledgment.

There is another problem with IDNDDE which DDClient also solves. IDNDDE may send data on an advise loop before it has sent an acknowledgement to the client ccepting the request. The client is not expecting data at that point so does not acknowledge it, stopping IDNDDE from sending any more data. DDClient has a mechanism for handling this early advise data.

[Application DDE interface topics](#) [General DDE FAQ topics](#) [DDE Software downloads](#) [Home page](#)



Ads by Google			
.Net XLS component Most Features, Performance, Support Satisfaction Guaranteed! 97-2007 www.activexls.com/Trial_Download/	Excel Server 2008 DIY your ERP, CRM, and MIS using Excel and SQL Server. It's easy! www.QinZhe.com/english/index.htm	PDF Writer for VB Create, display, print, edit, merge Royalty-free distribution. Try now! www.synactis.com	PowerBasic vs VisualBasic Faster. No Run-Times. No Bloat! CGI, Macros, ASM, Reg Expressions www.powerbasic.com

Site Sponsors

