

Integrating Visual Basic with Word for Windows

Presented by: **Woody Leonhard**

*Woody is president of Pinecliffe International, purveyors of the critically acclaimed **WOPR** — Woody's Office POWER Pack™ — a Windows Magazine Win100 Award laureate for best Windows product of 1993.*

*Woody wrote **Windows 3.1 Programming for Mere Mortals**, the seminal guide to Windows programming in Visual Basic, WordBasic, and other “macro” languages. Much of what follows is derived from discoveries detailed **Mere Mortals**.*

*Woody and Vincent Chen teamed up for the ultimate, no-bull reference to things WinWord, the **Hacker's Guide to Word for Windows**.*

*Pinecliffe International
Coal Creek Canyon
Golden, Colorado USA 80403-0100*

***WOPR** order line: 800-OK-WINWORD (800-659-4696)
Outside USA: 314-965-5630. “Takes WinWord places it's never been before.” — Windows User Magazine 2/93*

Windows 3.1 Programming for Mere Mortals, Addison-Wesley, 1992, ISBN 0-201-60832-4. “Manic and inspired.. my kind of writer.” — *PC Techniques* 11/92

Hacker’s Guide to Word for Windows, Addison-Wesley, 1993, ISBN 0-201-63273-X. “The best computer reference book I’ve ever seen.” — Scott Krueger, *Hackus Maximus*.

Best of Both Worlds

Whatever is taught liberally by your own teacher,

You have to deliver back to the Worthy.

— Zarathustra, *Fourth Admonition*, ca 660 BC

Let's face it: WinWord is not terribly well endowed when it comes to interaction with a user. Yes, you can have a Message Box pop out and ask Abort/Retry/Ignore? Yes, you can ask the user to type in things with an Input Box — but heaven help ya if the user hits Enter to start a new line. Yep, you can put together a custom dialog box — as long as it's pretty small and not terribly intelligent.

But that's about the extent of it.

Contrariwise, Visual Basic is a world-class underachiever when it comes to producing documents. You can print out a page or two, maybe change fonts around and the like, but if you want your output pretty or fancy, Visual Basic will leave you pulling your hair out.

Wait! Those two glasses aren't half empty! *They're more than half full.....*

Look at it the other way around: Visual Basic is the most sophisticated user interface engine on the face of the planet; and WinWord gives you so many options for creating fancy output that it'll make your head swim.

How to take advantage of each products' strengths? Easy. Dynamic Data Exchange.

DDE

We won't have enough time to beat the topic to death, so I'll take the Cliff Notes approach to Dynamic Data Exchange. There are many nuances and complications (much of *Windows 3.1 Programming for Mere Mortals* is consumed with the details), but for now I'll just hit the high points. Check out Chapter 6 of *Mere Mortals* for a much more thorough explanation.

DDE is a way for two Windows applications to talk to each other. A DDE conversation is just like a telephone call. The application placing the call — the app in the driver's seat — is the "Client"; the application answering the phone is the "Server".

A DDE conversation goes like this:

- The Client looks up a phone number, picks a topic of conversation, and dials the phone.
- If the Server is home (i.e., if it's running), and it likes the topic of conversation, it answers the phone.
- The Client asks the Server to do its bidding.
- The Server acknowledges that it has completed the task.
- The Client may ask the Server to do its bidding again, over and over, and may issue as many commands (pertaining to the current topic of conversation) as it likes.
- Finally, if they're polite, either the Client or the Server hangs up the phone. (If they're extraordinarily rude, one or the other may just walk away from the phone and leave the line open!)

That's the whole DDE shtick.

Programming it is another story, of course.

We're going to build a couple of DDE programs here in a second, then tie them together to make something worthwhile.

As you slog through the code, keep one thing in mind: if you tried to do this in C (as opposed to WordBasic and Visual Basic), you'd be looking at a week-long project with many hundreds of lines of code, and the near certitude that your DDE application, once complete, will be riddled with bugs.

Poker

Let's put together a little application that uses WinWord and Visual Basic at what they do best. I call it Poker.

Let's say you're working with a rather sophisticated Word for Windows document, one with several bookmarks in various places, scattered throughout the document.

The Poker approach (which you'd no doubt customize for your particular situation) is a supercharged variant on the WinWord {fillin} field. Run Poker with a WinWord document open, and you'll be presented with a list of all the bookmarks in the document. Pick a bookmark, type in a value, and *poof!* the stuff you typed is inserted at the bookmark location.

Simple, but potentially very powerful.

George Mair did something similar to automate the generation of legal documents. We built an invoicing and registration system around a Poker-like engine. You might want to use a Poker-style approach instead of the relatively brain-dead WinWord {fillin} field, to give your users a sophisticated way to fill out forms.

WinWord as Client

If you're trying this in front of a computer, start by cranking up both WinWord and Visual Basic.

We're going to write a little routine that sends Visual Basic a list of all the bookmarks in the current document.

While there are more sophisticated ways of accomplishing the same thing (in fact, you can have Visual Basic do all the work!), it's much simpler to demonstrate what's happening by starting with a simple WordBasic macro. I'll call it ... um ... Poker.

Preparing Visual Basic

Go into Visual Basic and start with a new form. Enable Dynamic Data Exchange to that form by clicking on LinkMode in the Properties box, changing it a 1 — Source. (You have to explicitly enable DDE, or VB won't permit it. Tough buzzard!)

VB should look like this:

Now double-click on the Text Box control, the box over on the left that says "ab". Move it around a little bit on the form, so it looks something like this:

Stick a couple of Label Controls on POKER's form (double-click on the capital "A", then smooch the box around). Add one ListBox control (double-click on the thing that looks like a list box in the fifth row of icons), and two push buttons (double-click on the oval thingie). The exact size and location isn't important.

Type in a few captions and you should have a form that looks something like this:

*We're using VB 2.0,
Standard Edition, but
you can use VB 1.0 or
VB Pro; they all work
about the same way.*

POKER DDE Handler

Next, we need to teach POKER how to respond to DDE Execute commands. Visual Basic is funny that way. WordBasic will respond to (almost) any macro or built-in function command, when it's sent over the phone.

But Visual Basic has to be taught precisely how to respond to each command: nothing is assumed, everything is hand-built.

We'll be sending POKER two different kinds of commands. The first kind tells POKER the name of the current WinWord document. We'll call that a Doc command, and make it look like this:

```
Doc c:\winword\whatsup.doc
```

(We could make it look like anything at all; there's nothing magic about the commands, or what they look like. It's Zig-Zag programming: strictly roll-your-own, doncha know.)

The second kind of command tells POKER to add a bookmark to its list of bookmarks. As I wasn't feeling particularly inspired when I wrote this, I settled on a drab command that looks like this:

```
Add TheNameOfTheBookmark
```

Double-click on the form, and flip down to the LinkExecute procedure. That's where you teach Visual Basic how to respond to DDE Execute commands. Like this:

That's not as complicated as it looks. Let's take it one line at a time.

Cancel is a special variable you can use to tell the DDE Client (in this case, WordBasic) that the DDE Execute statement screwed up. We know everything will go right (*right?*), so I just turned Cancel off.

If the first four characters of the DDE Execute command are "Doc ", I want to store away the name of the WinWord document. If the first four characters

*Visual Basic will be
able to tell a Doc
command from an
Add command by
simply looking at the
first four characters
of the command:
they're either "Doc "
or "Add " Easy, eh?*

are “Add ”, I want to add the bookmark name to the list. That’s all it takes. Two little things: I erased the “Text1” text in the text box. And I put a Dim WinWordDocName\$ statement in the declarations section, so that variable will be available to all the subroutines in POKER.

Time to save. Click on File, then Save Project. You can name the Poker form anything you like. When VB asks you for a name for the .MAK file, type in POKER. There. You have the beginnings of a VB application (or “project”) called POKER.

Finally, turn your nascent program into something real by clicking on File, then Make EXE. Name the program POKER.EXE, and make a mental note where you put it.

WinWord Calls

We’ll need a real short WordBasic macro to send bookmarks winging to the Visual Basic program POKER. Let’s call the macro SendBookmarks. Click on Tools, then Macro, make sure the Global Commands button is clicked, type in SendBookmarks, and enter the program. Something like this:

```
Sub MAIN
Shell "c:\vb\poker"
cn = DDEInitiate("poker", "Form1")
DDEExecute cn, "Doc " + FileName$()
For i = 1 To CountBookmarks()
    DDEExecute cn, "Add " + BookmarkName$(i)
Next i
DDETerminate cn
End Sub
```

Let’s go through it slowly.

The Shell statement should point to wherever you put POKER.EXE. (Yes, there are much fancier ways of doing all of this, but for now this will get us going.)

WordBasic initiates a DDE conversation by looking for an application called POKER, and proposing a topic of conversation called “Form1”. If Windows can find POKER, and it responds to a DDE telephone call based on the topic “Form1”, the DDE link will be initiated, and something called a “channel number” is returned to WordBasic.

Sure enough, POKER is sitting around. It has a form called “Form1”. So, if all goes well, the DDE conversation starts.

Then our WordBasic program DDE Executes the “Doc ” command, and as many “Add ” commands as are necessary to get all the bookmarks. Finally, the WordBasic macro hangs up the phone, terminating the DDE conversation.

Putting it Together

POKER is now half-done. *(Some would say half-baked, eh?)*

We now have the wherewithal to send a list of valid bookmark names to POKER, and leave POKER hanging around, waiting for the user to pick a bookmark and type in new text for that bookmark.

Bring up a WinWord document with a couple of bookmarks in it. Then run SendBookmarks. See that?

There. That wasn't so hard, was it? You've just completed a rather sophisticated DDE conversation with WinWord as Client, VB as Server.

VB as Client

Now it's Visual Basic's turn.

We need to set up the POKER form so it will send stuff back to WinWord. And it ain't as tough as you think.

Yell if this is getting too difficult, pilgrim. It's going by at autobahn speed; Mere Mortals takes a more leisurely pace. Sometimes you gotta stop and scarf the pizza....

We've seen some strange things in DDEland, but this is one of the strangest: POKER doesn't have to be running for the DDE conversation to be initiated! Usually. Lee Hudspeth was first to point that out to me. Most of the time, it's good enough if POKER is just open!

Halt!

Double-click on the push button that says “Cancel”. Type in the one line “End”. Like this:

That just tells POKER to stop if the user clicks Cancel.

The Brains

Time to write the program that actually updates the WinWord document, slamming the specified New Text into the chosen Bookmark location.

We’re going to do something that may seem a bit ... uh ... counter-intuitive.

While everything up to this point has concentrated on WordBasic being in the driver’s seat, acting as the DDE Client, and POKER acting as the DDE Server, it’s a whole bunch easier to update the WinWord document by starting a *second* DDE conversation, with POKER acting as DDE Client, and WinWord acting as Server.

If you dig into DDE, you’ll discover that it’s often easier to turn a problem inside-out, look at it from a different perspective. This is one of those cases.

While WinWord and SendBookmarks get the ball rolling, they kind of leave everything in never-never land. The user is confronted with a dialog box. If they pick a bookmark, optionally type in new text, and click “Update WinWord Document”, POKER takes things into its own hands, re-initiating contact with WinWord, changing the text in the specified bookmark, then croaking gracefully.

Neat, eh?

Double-click on the push button that says “Update WinWord Document”. Then type in this program:

Translating that into English, POKER picks up the phone and tries to contact WinWord, using the active document as a topic. The LinkMode command sets up a “cold” link (much more about hot, cold and warm links in *Mere Mortals*).

The LinkItem command picks up the bookmark selected by the user. Then POKER DDE pokes the contents of the Text1 text box into the selected bookmark. Finally, POKER hangs up the phone and dies.

The Result

Save everything. Click on File, then Make .EXE file.

Bring up a WinWord document (preferably with a few bookmarks in it), and start SendBookmarks. See how that works?

SendBookmarks wakes up POKER. It sends POKER the name of the current document (so POKER can re-establish the DDE link with the right document), then one-at-a-time sends all the valid bookmark names over to POKER.

When SendBookmarks is done, it leaves POKER up on top, ready for the user to make their choices.

The user can do whatever they like: everything is stable at this point; if they cancel out POKER, for example, there's nothing detrimental that will happen. It's stable.

Should the user pick a bookmark and click on "Update WinWord Document", POKER re-establishes contact with WinWord, picks the correct document, and the entire contents of the Text1 text box are shuffled to the indicated bookmark.

So What?

Well, yes, it's true that you could mimic most of POKER in WordBasic; the only thing you can't do in "plain vanilla" WordBasic is create a so-called "modeless" dialog box — one that just sits there, without monopolizing WinWord, forcing the user to satisfy the dialog box before they do anything else.

But this only scratches the surface of what you *could* do with a POKER-like approach.

The entire Visual Basic repertoire is at your fingertips. Database access. (*Access access, for that matter!*) OLE. Dialog boxes that are smart, that can sense when and where the mouse has gone, that can get at things like the Windows Common Dialog routines, menus, multiple-window interfaces, the whole nine yards.

It's only a DDE away.

What's Next?

I hope this little demonstration has given you an idea of how powerful DDE

can be, particularly when it's used to tie together two applications as different as WinWord and Visual Basic.

The possibilities are astounding.

There was an interesting demonstration of how Visual Basic can be used as a front end to 1-2-3/Windows, in an article by Justin Fielding in the April 1992 issue of *Lotus Magazine*.

In fact, you can use Visual Basic as a stunning user interface for *any* Windows application that supports DDE. The ... basic ... idea is quite simple: shell out to your custom Visual Basic program from inside the app, then use DDE to bring back the bacon.

“The idea is to make super-applications by writing macros that control and tie together commercial applications like Word and Excel. This is a sleeper concept that will be very big in the next few years.”

— *PC Techniques*, 11/92

“Visual Basic is a macro language.”

— Woody Leonhard, 3/92

Staying Connected

If you're just starting out with DDE, it would behoove you to pick up one of the books that deal with the subject.

The stuff you've seen here, though, is pretty much state-of-the-art, albeit in abbreviated form. Times are changing quickly.

How to find out about all these strange hacks? Easy. Get on line!

The best way to get a straight answer to your questions — from just-getting-started novice to grizzled guru — is in the Microsoft fora on CompuServe. If you aren't on-line, you're missing the greatest resource of WinWord help in existence.

General WinWord questions are handled on the Word forum: just type GO MSWORD. Questions specific to WordBasic and how it interacts with the outside world (including Windows API calls and DDE) run through the “Programming Microsoft Applications” forum. Type GO PROGMSA.

And Visual Basic topics are in the GO MSBASIC forum.

See you on-line!

Being fond of System is better than merely knowing it.

Taking one's delight in it is better than merely being fond of it.

— Confucius, ca. 500 BC

*Woody Leonhard
Coal Creek Canyon, Colorado
January 15, 1992*