VFP Advanced: Is This the Next Visual FoxPro?

Eric Selje
Salty Dog Solutions, LLC
Madison, WI USA
Voice: 608-213-9567
Website: www.SaltyDogLLC.com
Email: Eric@SaltyDogLLC.com

Did you know there is (kind of) a Visual FoxPro version 10? VFP Advanced (VFPA) fixes over 80 bugs in Visual FoxPro 9, introduces a couple of new features and data types, and also (drumroll, please) comes in a 64-bit version!

In this session, we'll dive into this free patch for Visual FoxPro. We'll go over some (but not all 81, as of this writing) bugs that it fixes, discuss what being 64-bit does for us, and have a conversation about whether you want to develop and release software that uses this patch. Is it dangerous? Buggy? Well-supported? Let’s find out…

What is VFP Advanced?

Imagine there's a group of employees at Microsoft. They are heartbroken when they hear "the announcement:" development on the Fox product line is ceasing. They know how great Visual FoxPro is and see the potential for what it can be. These employees are also aware of a few niggling bugs in the product and are determined to see them fixed and released to their devoted user base. And they have access to the Visual FoxPro 9 source code.

So, they go rogue. They start secretly releasing updates to Visual FoxPro 9 that fix over 80 bugs they found. And then they add a few features too. And with some brilliant coding, they even release the oft-requested 64-bit version of Visual FoxPro 9 as well!

Those releases are Visual FoxPro Advanced.

But these releases aren't from a team of people, it's only one man, a Mr. Chuanbing Chen from Shenzen, China. And Chen does not nor never has worked for Microsoft. So, Chen does not even have access to the Visual FoxPro 9 source code.

Yet somehow, since 2012, Chen is publicly releasing "updates" to VFP9 as a product called Visual FoxPro Advanced.

How does Chen do this?

It may seem like an impossible task, since there is no source code. Chen doesn't even have access to Intermediate Language source code he could get from decompiling a .Net application.

According the Chen himself, it is indeed "very difficult" because he only has access to the compiled binaries, the same as you and I. When a need for a change is identified, he painstakingly wades through the binary code to identify the area that needs modifying, codes and tests his changes, and then writes the altered binary back out to a new binary. I personally can't imagine undertaking this endeavor, yet Chen has done it. "Very difficult?" I think Chen is modest and understated. See *Appendix B* for a sample of his work.

Is this safe?

A valid question. We don't know what changes he's made to the development environment or the new runtimes. Could Chen have added code that subversively ships data out the door without our knowing it? This would be a fairly easy thing to check by enabling the logs on your firewall and watching where data goes when your application is running. If it turned out to be happening, his reputation would be immediately ruined once word got out.

My anti-virus software did actually flag the download as "potentially unsafe," but that was because it had no "reputation" yet. So few people have downloaded VFPA that Symantec couldn't assure me that it was safe.

He could also have inadvertently introduced a bug to Visual FoxPro that wasn't there before, but according to Chen he has over 600 people that use and test Visual FoxPro

Advanced, so if something was amiss he would hear about it. One of his main customers is Chen himself. He has written management systems in VFPA that are used by many customers in restaurants, factories, and other businesses.

Another customer is Frank Moore of FWM Software in California (http://www.fwmsoftware.com). Mr. Moore uses VFPA to develop and deploy his commercial software application, LienWriter. He has told me he has had virtually zero issues with VFPA, and indeed its bug fixes and new features have been invaluable to him.

But as it says at the bottom of each of Chen's fix reports, "There is no guarantees."

Is Microsoft cool with this?

According to Chen, Microsoft has been aware of what he's doing for quite a while. Their official response is "We appreciate you reaching out to Microsoft about this application. However, we cannot comment on legality because it was created by a separate entity that is unaffiliated with Microsoft. I recommend continuing to proceed with caution and perhaps request that the company provide you with proof of legal ownership as well as legality of doing business with US customers." I'm no lawyer, but the "proceed with caution" leaves the door open for Microsoft to put their foot down if they wanted to, but at this point they've got bigger fish to fry.

As extra precautions against legal issues, Chen does not change any copyright information anywhere in the product as well, and he's selling Visual FoxPro Advanced for exactly ¥0.00 (which as of this writing converts to USD $0.00). He would like anyone who uses it to subscribe to support and services for a whopping ¥10,629.50 (or USD $100) per year.

How do I get VFP Advanced?

All right so you're convinced to give VFPA (a clever acronym, as we know A is hex for '10') a try. This first thing you must do is have a licensed copy of Visual FoxPro 9 on your system and have it patched to version 9.0.0.7423. If you don't, VFPA's setup routine will tell you that you haven't met the prerequisites and fail.

Table 1: VFPA Prerequisites

File name	Version	Date	Size
VFP9 IDE			
C:\Program Files (x86)\Microsoft Visual Foxpro 9\Vfp9.exe	**9.0.0.7423**	**2009-4-3**	**5,783,552**
VFP9 IDE language resource file			
C:\Program Files (x86)\Microsoft Visual Foxpro 9\Vfp9enu.dll	**9.0.0.5815**	**2007-10-15**	**1,507,328**
VFP9 Runtime			
C:\Program Files (x86)\Common Files\Microsoft Shared\VFP\Vfp9r.dll	**9.0.0.7423**	**2009-4-3**	**4,734,976**
VFP9 Multi-threaded Runtime			
C:\Program Files (x86)\Common Files\Microsoft Shared\VFP\Vfp9t.dll	**9.0.0.7423**	**2009-4-3**	**3,907,584**
VFP9 Runtime language resource file			
C:\Program Files (x86)\Common Files\Microsoft Shared\VFP\Vfp9renu.dll	**9.0.0.5815**	**2007-10-15**	**1,187,840**

If you don't have your Visual FoxPro 9 completely patched up to service pack 2, hotfix 3 already, you can get Service Pack 2 from https://docs.microsoft.com/en-us/previous-versions/visualstudio/foxpro/mt490117(v=msdn.10) and HotFix 3 from https://github.com/VFPX/VFP9SP2Hotfix3. They are also conveniently available on Chen's website.

To get VFPA itself, go to Chen's website at http://www.baiyujia.com/vfpadvanced/f_vfpa_about.asp. Near the bottom are the download links for both the 32 and 64 bit versions.

Installing is very straightforward. You can install either or both of these versions of VFPA right alongside your existing Visual FoxPro 9. The installer doesn't do anything at all to your original VFP installation folder(s).

After VFPA is fired up for the first time, the experience is so … much the exact same as the first time VFP9 started, right down to the Task Pane Manager **(Figure 1)** that you will immediately disable from startup. But a quick check of the About screen **(Figure 2)** reveals that this is indeed the updated version.

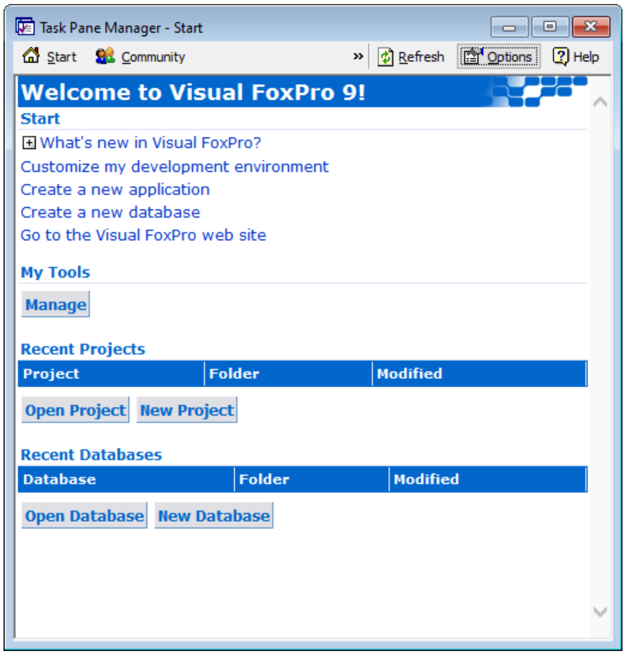
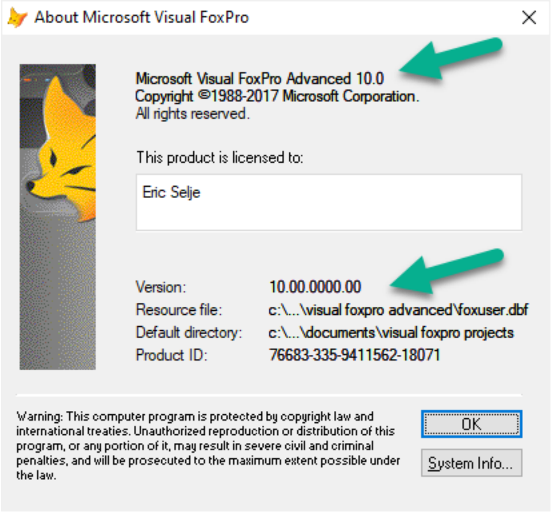


Figure 1: Remember Me?


About Microsoft Visual FoxPro

Microsoft Visual FoxPro Advanced 10.0
Copyright ©1988-2017 Microsoft Corporation.
All rights reserved.

This product is licensed to:

Eric Selje

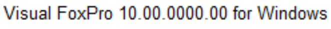
Version: 10.00.0000.00
Resource file: c:\...\visual foxpro advanced\foxuser.dbf
Default directory: c:\...\documents\visual foxpro projects
Product ID: 76683-335-9411562-18071

Warning: This computer program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Figure 2: Version 10!

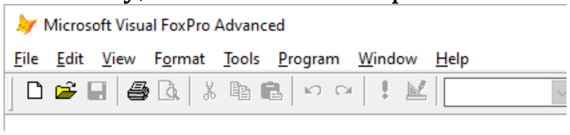
Just to doublecheck though, let's try

```
? version()
```



```
Visual FoxPro 10.00.0000.00 for Windows
```

And lastly, notice the new caption at the top:

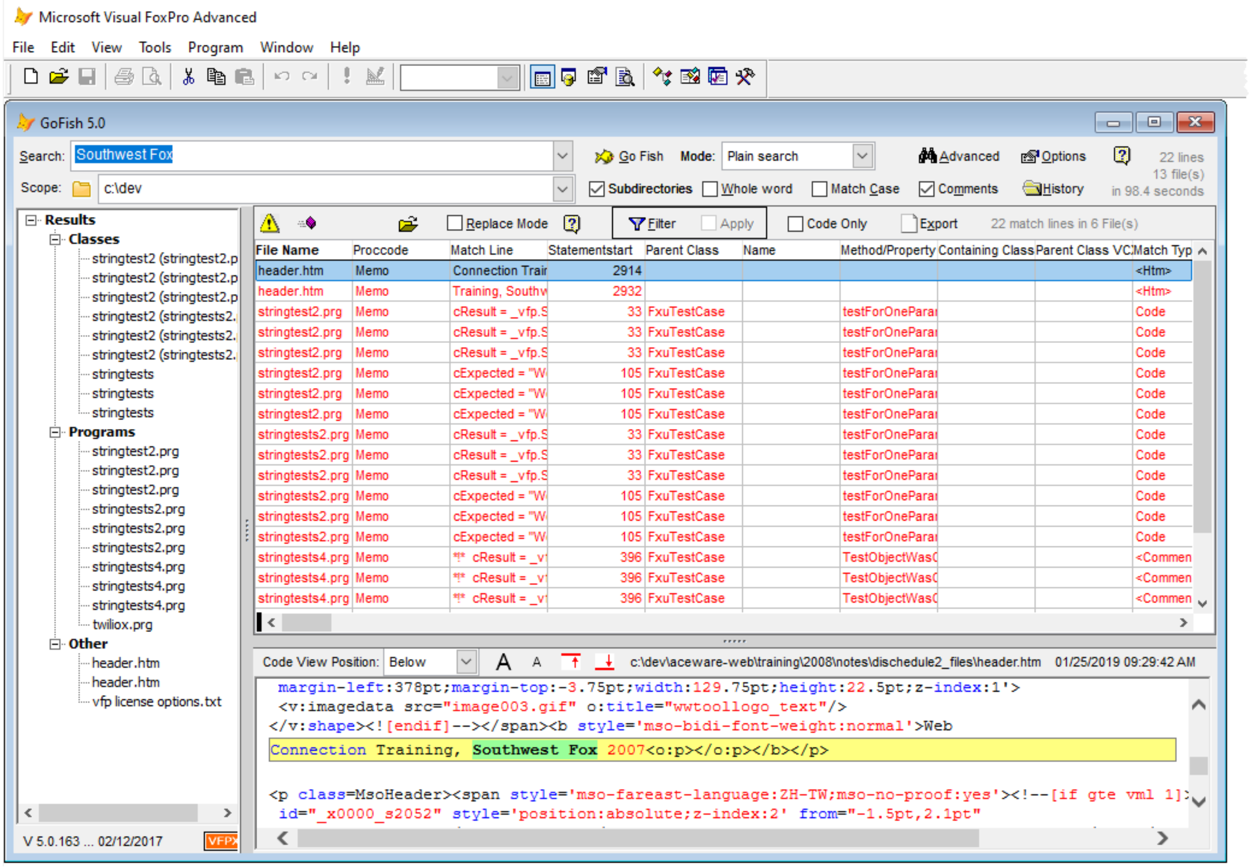


Already Chen's done more than I would have been able to, but there's so much more!

Does it Work?

After verifying that I was indeed running VFPA and not VFP9, I poked around a bit. Everything looked exactly the same as VFP9, and that's because (and this is important to keep in mind), it is the same. VFPA is VFP9, but with some internal changes. The UI, designers, and compiler are the same. The runtime is merely a patch of the existing VFP9 runtimes, in order to support the new features, although it does do something interesting: it incorporates all of the separate language files into the VFPA runtime. This reduces the number of files you may need to ship with your application but increases its size. I'd show you a screenshot to emphasize the point, but you would just say "That looks just like VFP9."

The next thing I attempted to do once I started VFPA is to see if it was compatible with existing applications. I started with something I use every day: GoFish5. This is a good test because GoFish uses low level functions, ActiveX controls, and local DBF tables. It worked perfectly.



Search: Southwest Fox

Scope: c:\dev

Mode: Plain search

☑ Subdirectories ☐ Whole word ☐ Match Case ☑ Comments

22 lines
13 file(s)
in 98.4 seconds

☐ Code Only

22 match lines in 6 File(s)

- Results
  - Classes
    - stringtest2 (stringtest2.p
    - stringtest2 (stringtest2.p
    - stringtest2 (stringtest2.p
    - stringtest2 (stringtests2.
    - stringtest2 (stringtests2.
    - stringtest2 (stringtests2.
    - stringtests
    - stringtests
    - stringtests
  - Programs
    - stringtest2.prg
    - stringtest2.prg
    - stringtest2.prg
    - stringtests2.prg
    - stringtests2.prg
    - stringtests2.prg
    - stringtests4.prg
    - stringtests4.prg
    - stringtests4.prg
    - twiliox.prg
  - Other
    - header.htm
    - header.htm
    - vfp license options.txt

File Name	Proccode	Match Line	Statementstart	Parent Class	Name	Method/Property	Containing Class	Parent Class VC	Match Typ
header.htm	Memo	Connection Trair	2914						<Htm>
header.htm	Memo	Training, Southw	2932						<Htm>
stringtest2.prg	Memo	cResult = _vfp.S	33	FxuTestCase		testForOnePara			Code
stringtest2.prg	Memo	cResult = _vfp.S	33	FxuTestCase		testForOnePara			Code
stringtest2.prg	Memo	cResult = _vfp.S	33	FxuTestCase		testForOnePara			Code
stringtest2.prg	Memo	cExpected = "W	105	FxuTestCase		testForOnePara			Code
stringtest2.prg	Memo	cExpected = "W	105	FxuTestCase		testForOnePara			Code
stringtest2.prg	Memo	cExpected = "W	105	FxuTestCase		testForOnePara			Code
stringtests2.prg	Memo	cResult = _vfp.S	33	FxuTestCase		testForOnePara			Code
stringtests2.prg	Memo	cResult = _vfp.S	33	FxuTestCase		testForOnePara			Code
stringtests2.prg	Memo	cResult = _vfp.S	33	FxuTestCase		testForOnePara			Code
stringtests2.prg	Memo	cExpected = "W	105	FxuTestCase		testForOnePara			Code
stringtests2.prg	Memo	cExpected = "W	105	FxuTestCase		testForOnePara			Code
stringtests2.prg	Memo	cExpected = "W	105	FxuTestCase		testForOnePara			Code
stringtests4.prg	Memo	*!* cResult = _v	396	FxuTestCase		TestObjectWas(			<Commen
stringtests4.prg	Memo	*!* cResult = _v	396	FxuTestCase		TestObjectWas(			<Commen
stringtests4.prg	Memo	*!* cResult = _v	396	FxuTestCase		TestObjectWas(			<Commen

Code View Position: Below

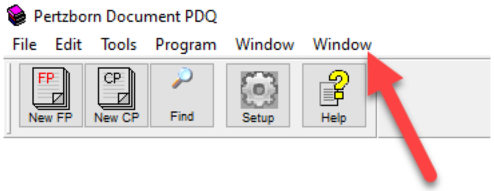
c:\dev\aceware-web\training\2008\notes\dischedule2_files\header.htm 01/25/2019 09:29:42 AM

```
  margin-left:378pt;margin-top:-3.75pt;width:129.75pt;height:22.5pt;z-index:1'>
  <v:imagedata src="image003.gif" o:title="wwtoollogo_text"/>
</v:shape><![endif]--></span><b style='mso-bidi-font-weight:normal'>Web
Connection Training, Southwest Fox 2007<o:p></o:p></b></p>

<p class=MsoHeader><span style='mso-fareast-language:ZH-TW;mso-no-proof:yes'><!--[if gte vml 1]>
  id="_x0000_s2052" style='position:absolute;z-index:2' from="-1.5pt,2.1pt"
```

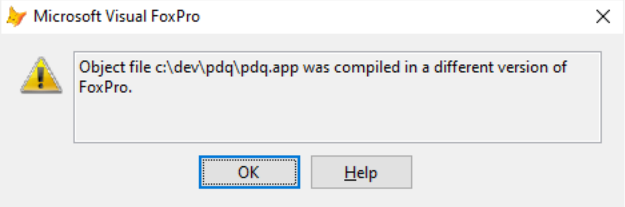
Next, I tried one of my own work applications, one that uses ODBC to talk to an Informix database. I can't show you a screenshot because the data is proprietary, but it worked perfectly.

I checked Thor, FoxUnit, FoxTabs, and Project Explorer: all worked perfectly.

Almost convinced, I checked another custom application of mine. On this one I wanted to actually make some changes, compile it, and run it. The entire experience was just like VFP9 (again, because it really is mostly VFP9). When I ran the application though I noticed a little quirk with the menu: there are two Window menus. Can that be right?



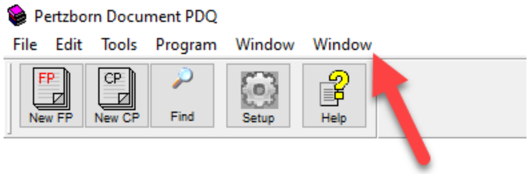
I fired up VFP9 to run the program to see if this problem existed back in VFP9 as well. When I ran the application it said:



Object file c:\dev\pdq\pdq.app was compiled in a different version of FoxPro.

Well of course it does – that makes sense. I had compiled the app in VFPA, so VFP9 won't recognize it. Fair enough. I recompiled it with VFP9 and ran it again:

It's still there! So VFPA even reproduces some of the quirks of VFP9. When I hover my mouse over the 2nd Window menu in either version, it changes to the proper Help menu. I'll have to dig into this further, but it doesn't appear to be an issue with VFPA itself.



What New Features and Fixes?

Satisfied that VFPA is at least as solid as VFP9, let's look at what it adds to the mix. The first time I heard that Visual FoxPro Advanced fixes over 80 bugs (and counting) in Visual FoxPro 9, I wondered how it is that I've haven't heard about all these issues. Well the answer is that Visual FoxPro 9 was a mature product that works for 99.99% of the people who use it. But there are quite a few esoteric issues and limitations with VFP9 that you may have even noticed depending on the nature of your application. And if you did hit one of these issues or limitations, it might be a deal breaker for your application!

In Appendix A, I've enumerated all the fixes and new features from Chen's website and expanded their descriptions. New features have a green bar next to them to identify them easily. Let's highlight a few of these.

SYS(9000)

I'm going to start with my favorite new command, SYS(9000). This command enables Fix 18, which disables the "shadow" of dockable windows such as the Command Window. In Windows 10, and especially on my 4k monitor, this shadow either lagged far behind where I was dragging or had no relationship at all to my mouse movement. This drove me nuts, and now it's fixed, and that would be enough for me right there to use VFPA.

OS()

Let's assume your company has moved to Windows 10 (and I really hope it has, because Windows 7 extended support ends in January, 2020), and you want to add features to your application that only work with Windows 10, like sending "toast" notifications to the Action Center.

In VFP9, there was no way to distinguish which version of Windows you were running beyond 7. OS() returns *Windows 6.02* regardless if you were running Windows 7, 8, or 10 because there was no version 8 or 10 the last time VFP9 was officially patched. VFPA fixes that by returning *Windows 10.00*. Now you can have code that acts differently.

Now you may have code that will actually break if OS() doesn't return "Windows 6.02". Chen has thoughtfully added a bunch of new SYS(90##) functions that will enable or disable many of the fixes that he's made, so your application can retain backward compatibility if necessary. For OS(), SYS(9019,0) will disable the new functionality, causing it to return the same thing it would in VFP9. Other SYS(90##) functions follow the same pattern.

Bitwise Calculations

Suppose you have a field in your table that uses bits as flags rather than separating them into separate logical fields (a fairly common thing). VFPA adds bitwise operators to the CALCULATE command, which is much easier than using MOD() or other methods to determine if a flag is set for any of your records (see Code Block 2).

```
CREATE TABLE userrights FREE (userid i, rights i)
* 0 = Read Only Accounts Receivable
* 1 = Can write to the Accounts Receivable
* 2 = Read Only Accounts Payable
* 4 = Can write to Accounts Payable
* 8 = Read Only Payroll
* 16 = Can write to Payroll
* ...
* 128 = Admin
* 256 = SuperUser

* Insert some users
INSERT INTO userRights (USERID, rights) VALUES (1, 0) && 00000000
INSERT INTO userRights (userid, rights) VALUES (2, 1) && 00000001
INSERT INTO userRights (userid, rights) VALUES (3, 23)      && 00010111
INSERT INTO userRights (userid, rights) VALUES (4, 56)  && 00111000
INSERT INTO userRights (userid, rights) VALUES (5, 34)  && 00100010
INSERT INTO userRights (userid, rights) VALUES (6, 148) && 10010100
INSERT INTO userRights (userid, rights) VALUES (7, 128) && 10000000
INSERT INTO userRights (userid, rights) VALUES (8, 65)      && 01000001
INSERT INTO userRights (userid, rights) VALUES (9, 52)      && 00110100

* We can now do this to determine who can do what...
CALCULATE BAND(rights), BOR(rights), BXOR(rights) TO bAnd, bOr, bXOr
? m.bAnd && Which rights does everybody have?
? m.bOr      && Which rights does nobody have?
? m.bXOr && Which rights does an odd number of people have?
* 109 = 64+32+8+4+1
```

Code Block 1: New operators for CALCULATE

VFPA Fixes Bugs

Have you ever been debugging your program and the debugger stops at the Trace window on a line of code that doesn't have a breakpoint? Well I have. It shakes your confidence in VFP9 a bit. But VFPA takes care of that problem and restores that confidence.

There must be something hinky when STATUS BAR=OFF and TALK=ON in VFP9, because it could crash your application if you used Varbinary or Blob Data (Fix 7) or Access and Assign methods (Fix 10). Those have been fixed in VFPA.

ON ERROR drops _TEXT

Here's a bug that was introduced in VFP9 that has the potential to break your program: VFP closes the file handle _TEXT before entering an ON ERROR routine. Let's see why this could be devastating with this example code:

```
ON ERROR DO MyErrorHandler WITH MESS(), MESS(1), LINENO()
cProcessLogName = FORCEEXT("ProcessLog v"+STR(VERSION(5),4),'txt')
iProcessLog = FCREATE(cProcessLogName)
_TEXT = iProcessLog
SET TEXTMERGE ON
* Send some text to our ProcessLog
\ <<DATETIME()>> Starting ProcessLog
\ <<DATETIME()>> Doing Thing 1...
* Uh oh, we have an error
ERROR 12
* Moving on...
\ <<DATETIME()>> Doing Very Important Thing 2. This better get logged!
\ <<DATETIME()>> Finished ProcessLog
FCLOSE(iProcessLog)
SET TEXTMERGE OFF
ON ERROR


PROCEDURE MyErrorHandler
LPARAMETERS cMessage, cCode, iLineNo
* Let's write that error to our ProcessLog
\ <<DATETIME()>> Error <<cMessage>> (<<cCode>>) at line <<iLineNo>>
ENDPROC
```

Code Block 2: There's a fatal flaw with this code in VFP9 that VFPA fixes

Looking at this code block, you wouldn't know there's a problem (besides the intentional error that I have introduced to demonstrate the bug). But there is. In VFP9, when ON ERROR fires, the file handle that _TEXT points to is cleared out! Any text merging from that point forward simply does not occur to the file, with zero indication to the user that that's happening.

Now depending on the nature of what you're writing to the file, this could be devastating. In my example, all logs beyond the error are not written out. If you were dynamically writing crucial code to a file to be executed, it would fail.

VFPA fixes this issue. And following its usual pattern, if you *want* this bug in your code you can use SYS(9023,1) to keep it in there. Thankfully VFPA defaults to SYS(9023,0) to fix the bug.

ROUND()

Here's an easy question: What's the value of ROUND(512.92,2)?

In VFP9, the answer is "It depends?" Check out this code block:

```
? ROUND(512.925000000,2)  && Displays 512.93
? ROUND(512.9250000000,2) && Displays 512.92
```

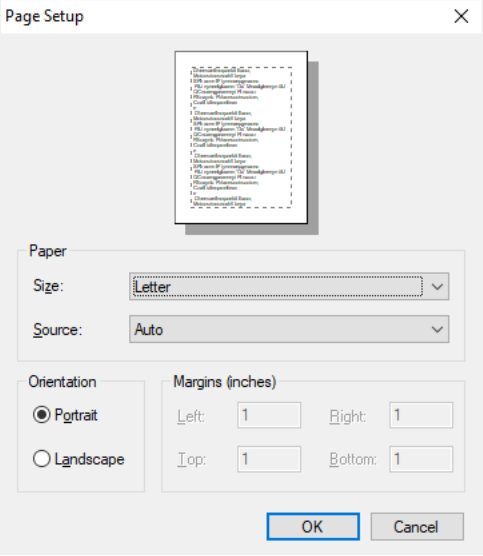
If you are doing precise calculations in VFP9 and rely on ROUND() being accurate, VFPA fixes it.

VFPA Straightens Out Quirks

There are a few things that Visual FoxPro 9 does that are…quirky. For example, if you use COPY FILE, the new file is always created with a lowercase filename. VFPA fixes COPY FILE so that it honors the case-sensitivity.

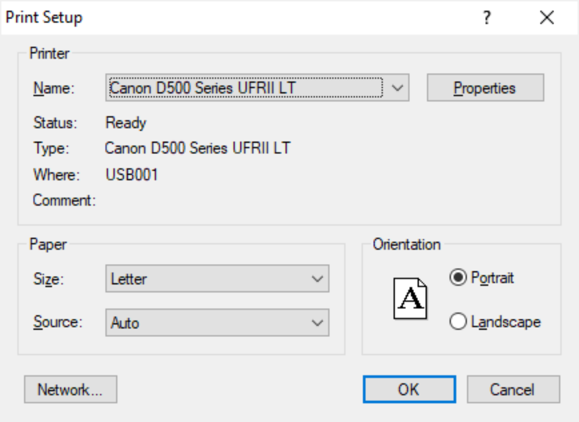
Private data sessions in VFP9 automatically revert to SET TALK ON and SET SAFETY ON, so the first thing you must do is set them to more sensible defaults. VFPA fixes that.

Somewhere between VFP8 and VFP9, Microsoft decided to remove the “Printer” selection from the “Page Setup” dialog in SYS(1037), which renders it somewhat useless. VFPA puts it back! *[If you’d like to see exactly how Chen fixed this, see Appendix B]*



Size: Letter
Source: Auto
● Portrait
○ Landscape
Left: 1 Right: 1
Top: 1 Bottom: 1

Figure 4: VFP9's Page Setup



Name: Canon D500 Series UFRII LT
Status: Ready
Type: Canon D500 Series UFRII LT
Where: USB001
Comment:
Size: Letter
Source: Auto
● Portrait
○ Landscape

Figure 3: VFPA's Page Setup - with Printer!

Anti-virus programs are often touchy when applications read or write files directly to disk. In VFP9, STRTOFILE(), FILETOSTR(), and BUILD EXE worked directly with disk (use the FileCreateA Windows API command), and this often got intercepted by anti-virus programs, causing Windows to pause while the action was inspected and validated. If you’re application moved on assuming the file was there, you may have been out of luck. VFPA directs those commands to use the Windows System Cache instead, which reduces the likelihood of anti-virus interference and has the pleasant side-effect of being quicker. You can revert these if you wish, but they will use the cache by default.

VFPA Overcomes Limitations

Did you know if a VFP9 form had a procedure that was greater than 64k, there was a chance it would crash your application? Regardless of the wisdom of having such a large form method, it was allowed so it was probably used. Someone discovered this bug, but VFPA quashes it.

There was a similar issue if you had a 64k or larger procedure in a program that also had macro expansion. Not anymore.

VFP9 has a limitation of 128 fields that you can name in a REPLACE command, but VFPA expands that to 255. Again I'm not going to question the wisdom of having that many fields in a REPLACE command, but at least you can do it now.

Conclusion Regarding 32-Bit Visual FoxPro Advanced

Visual FoxPro Advanced has fixed a lot more than I've even mentioned here. Quite a few of them have to do with quirks in the REPORT and Print Preview functionality of VFP9. A few fix bugs in the VFPOLEDB driver. Three of them allow you to configure how many times disk operations such as DELETE FILE, MKDIR and RMDIR are retried before failing. I'll leave it up to you to check out http://baiyujia.com/vfpdocuments/ and see the full (and ever-expanding) list of fixes that Chen has made.

After spending time with Visual FoxPro Advanced, I feel comfortable saying that if you're developing applications in Visual FoxPro 9, you should at least try Visual FoxPro Advanced. It won't hurt and may fix issues you don't even know you have. It's 100% compatible with VFP9 and a nice step forward, and has better official support than Visual FoxPro 9 does.

64-bit Visual FoxPro Advanced

Unbelievable as it may sound, Chen has *also* created a 64-bit version of Visual FoxPro Advanced. It compiles 64-bit apps, which of course will only run on 64-bit versions of Windows and aren't necessarily faster than 32-bit applications. So why would we want a 64-bit version?

Advantages of 64-bit VFPA

More RAM

One obvious advantage of a 64-bit version is that it can take advantage of more system RAM. A quick check of MEMORY() shows that it still shows 640, same as it has since since the earliest days of Fox, so that's worthless. SYS(1001), which "returns the virtual memory pool size, which is approximately four times the amount of physical memory" also shows the same value for both the 32 and 64 bit version. However, SYS(3050,1), which "returns the foreground buffer memory size" shows he same value as SYS(1001) in the 32-bit version of VFPA, but TWICE as much in the 64-bit version!

To allow additional memory usage, Chen uses something called "RIP-Relative Addressing" in his code. I'll leave it up to you to dig into that further if you're interested but it does get the job done.

Larger DBF Sizes?

You may be thinking that a 64-bit version of VFPA may finally allow us to overcome the 2 gigabyte limitation on DBF files. As of this writing that's not true, but Chen says he is working on this. At that file size, you may be wise to upsize your data anyway.

Better Runtime

VFP9 and VFPA (32-bit) both use the MSVCR71.dll C++ runtime behind the scenes. VFPA 64 uses the more recent and still supported MSVCR10.dll runtime. I could have sworn I saw an announcement in the last few days about Microsoft rolling all of their C++ runtimes into one package, but now I can't find it. If you find this information and share it with me before my session at Southwest Fox 2019, I'll buy you a beer!

64-Bit ActiveX Controls and ODBC Drivers

The most compelling case for using a 64-bit version of Visual FoxPro Advanced is that it allows you to use 64-bit ActiveX controls. For example, if your client has switched over to the 64-bit version of Microsoft Office, and your application does a lot of Office automation, you're out of luck unless your application is also 64-bit. Visual FoxPro 9 and even the 32-bit version of VFPA simply cannot use 64-bit ActiveX controls and COM objects, and there are some compelling 64-bit only controls, such as the ones from longtime Southwest Fox supports DBi Technologies (https://www.dbi-tech.com/).

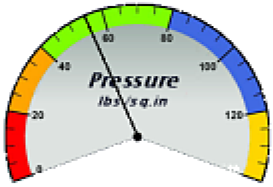
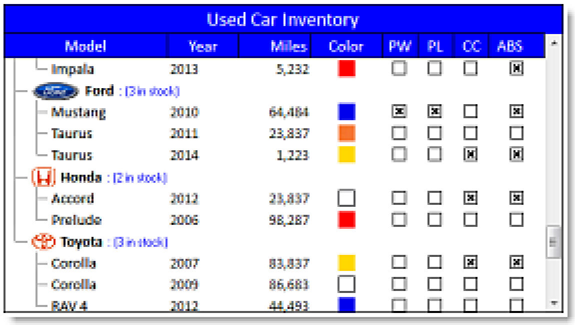


Figure 6: DBi's Gauge Control



Used Car Inventory

Model	Year	Miles	Color	PW	PL	CC	ABS
Impala	2013	5,232	🟥	□	□	□	☒
Ford : [3 in stock]							
Mustang	2010	64,484	🟦	☒	☒	□	☒
Taurus	2011	23,837	🟧	□	□	□	□
Taurus	2014	1,223	🟨	□	□	☒	☒
Honda : [2 in stock]							
Accord	2012	23,837	□	□	□	☒	☒
Prelude	2006	98,287	🟥	□	□	□	□
Toyota : [3 in stock]							
Corolla	2007	83,837	🟨	□	□	☒	☒
Corolla	2009	86,683	□	□	□	□	□
RAV 4	2012	44,493	🟦	□	□	□	□

Figure 5: DBi's TreeView Control

Similarly, you can take advantage of 64-bit ODBC drivers, with their larger capacity.

Disadvantages of 64-bit VFPA

32-Bit Controls and Drivers Do Not Work

If you have 32-bit ActiveX controls in your VFP application, they will not work and will need to be upgraded (if an upgrade can even be found). This includes some MDAC and Jet DAO drivers. The control this might affect most is the oleTreeView control that ships with MSCOMCTL32.ocx. There is no 64-bit equivalent, so you'll have to refactor your code to use the DBi Technologies TreeView control (**Figure 5**) . ListView, DateTimePicker, and other controls in that package will also be affected.

Is it Compatible with 32-bit VFPA?

Chen has done extensive work to make the 64-bit version of VFPA completely compatible with the 32-bit version. Any incompatibilities stem from circumstances beyond his control, like the ActiveX issues. He has extensively tested the DECLARE commands for using Windows APIs in VFP and claims most of them work, but not all of them.

When I tried running GoFish5 in the 64-bit version of VFPA it failed because GoFish5 uses TreeView and other 32-bit ActiveX controls. It will have to be refactored for 64-bit.

Chen has added a new _WIN64 constant that is only .T. for VFPA 64. This will allow for testing and conditional compilation of your applications so you won't have to maintain two separate codebases.

Conclusion Regarding 64-Bit Visual FoxPro Advanced

It's unknown how long vendors of 3rd party controls will continue to produce and support their 32-bit versions, or for that matter how long Microsoft will continue to put out 32-bit versions of Windows since most computers are 64-bit these days. Having a 64-bit option gives FoxPro developers the possibility of moving into an all 64-bit future, alleviating a grave concern. Because it's not a seamless transition, developers who want to ensure this future should begin testing their development tools (like GoFish) and their own applications in 64-bit VFPA to see what's possible.

Bonus: VFP C++ Compiler

Chen also sells another interesting product for $499 called *VFP C++ Compiler*. This product compiles VFP code into C++ code, and then recompiles that C++ code into a native Windows EXE. It works with VFPAdvanced (of course), but also all versions of VFP back to 6. He's been working in it since 2008 so it too is a mature product.

This product could have a complete session devoted just to it alone, but here are some of the highlights:

- There's an option to include vfpar.dll and vfpcore.dll right into the compiled EXE, so no additional runtimes (except the necessary Visual C++ runtimes) need to be distributed with your application.

- The compiled executable file and the dynamic link library file can be compressed with UPX or other compression tools. For the executable file, the compression rate is between 5% and 30%. For the dynamic link library file, the compression rate is between 30% and 50%.

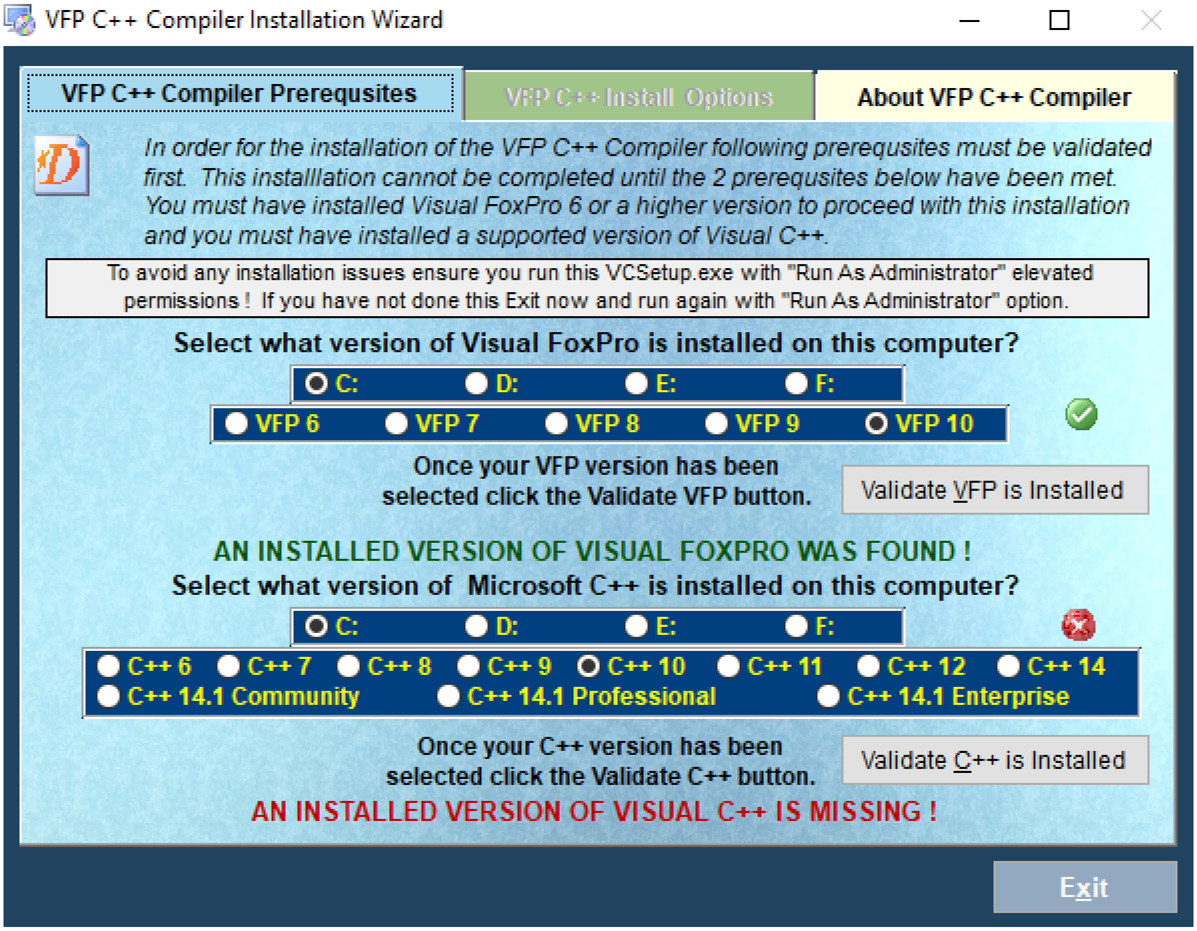
- Includes encryption options to make any attempt to decompile or analyze the generated file in a debugger a massive, time-consuming effort (and that's Chen saying that, who knows effort).

- Supports digital signing of the compiled EXE

- Compiles quickly with options for incremental compilation as well as multi-threaded compilation.

- The resulting application may (probably will) run faster because:
  - The VFP C++ Compiler use a global NTI (Name Table Index), so when it calls a program (compiled with VFP C++ Compiler) it does not need to build new NTIs since they already exist.
  - Usually, the VFP C++ Compiler will compile the code block (many lines) down to one function. When the function is called, it will not run the VFP idle loop until the function completes.
  - If the executable file compressed with the UPX program, it is smaller, so it will be loaded faster from disk.

Note that, like VFPA itself, the VFP C++ Compiler requires VFP9 to be installed **(Figure 7).** It also requires a Visual C++ compiler. I used the Community Version of Visual Studio 2015 (14 on the setup screen, natch). Newer versions are not yet supported.



VFP C++ Compiler Installation Wizard

In order for the installation of the VFP C++ Compiler following prerequsites must be validated first. This installlation cannot be completed until the 2 prerequisites below have been met. You must have installed Visual FoxPro 6 or a higher version to proceed with this installation and you must have installed a supported version of Visual C++.

To avoid any installation issues ensure you run this VCSetup.exe with "Run As Administrator" elevated permissions ! If you have not done this Exit now and run again with "Run As Administrator" option.

Select what version of Visual FoxPro is installed on this computer?

● C: ○ D: ○ E: ○ F:

○ VFP 6 ○ VFP 7 ○ VFP 8 ○ VFP 9 ● VFP 10 ✓

Once your VFP version has been selected click the Validate VFP button.

AN INSTALLED VERSION OF VISUAL FOXPRO WAS FOUND !

Select what version of Microsoft C++ is installed on this computer?

● C: ○ D: ○ E: ○ F: ✗

○ C++ 6 ○ C++ 7 ○ C++ 8 ○ C++ 9 ● C++ 10 ○ C++ 11 ○ C++ 12 ○ C++ 14
○ C++ 14.1 Community ○ C++ 14.1 Professional ○ C++ 14.1 Enterprise

Once your C++ version has been selected click the Validate C++ button.

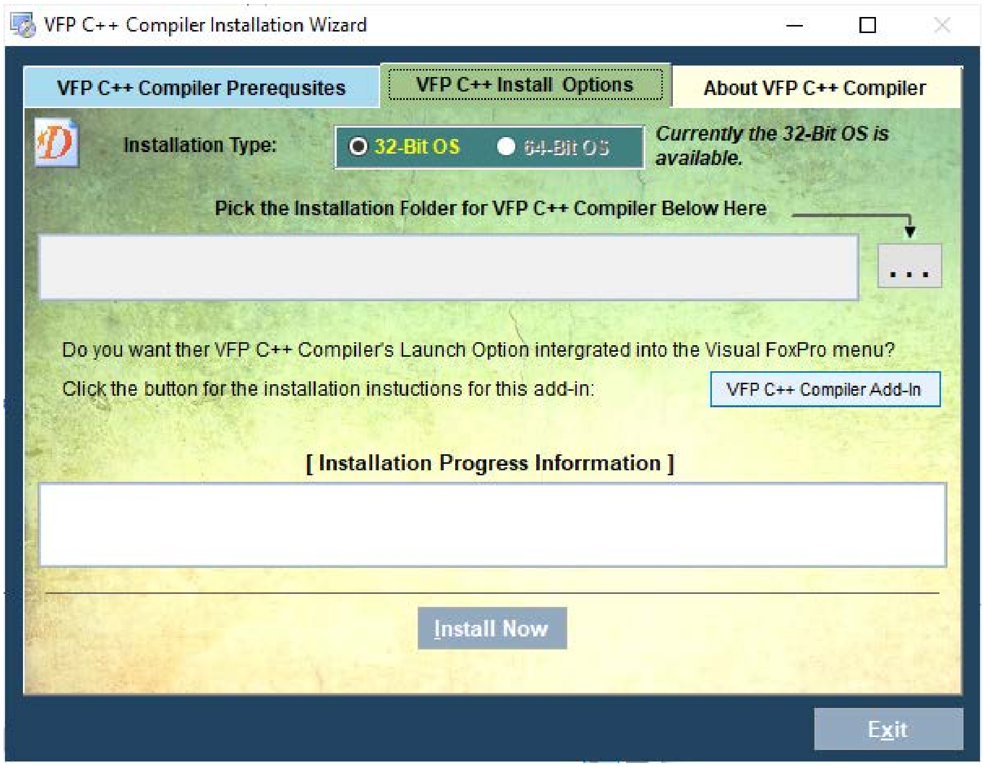
AN INSTALLED VERSION OF VISUAL C++ IS MISSING !

Figure 7: The VFP C++ Compiler Installation Wizard, Prerequisites

Once you've satisfied the prerequisites, you can view the Install Options **(Figure 8).**
Note that only 32-bit is available to me because I installed a 32-bit version of Visual Studio.

There’s even a way to add the C++ Compiler to the FoxPro Project menu if you’d if you follow the extra instructions (**Figure 9**).

During installation, detailed progress information is given to ensure you things are moving along. An interesting fact about the VFP C++ Compiler Installation Wizard is that it itself was written in VFP code and compiled with the VFP C++ compiler (**Figure 10**).



VFP C++ Compiler Installation Wizard

Installation Type: ● 32-Bit OS ○ 64-Bit OS

Currently the 32-Bit OS is available.

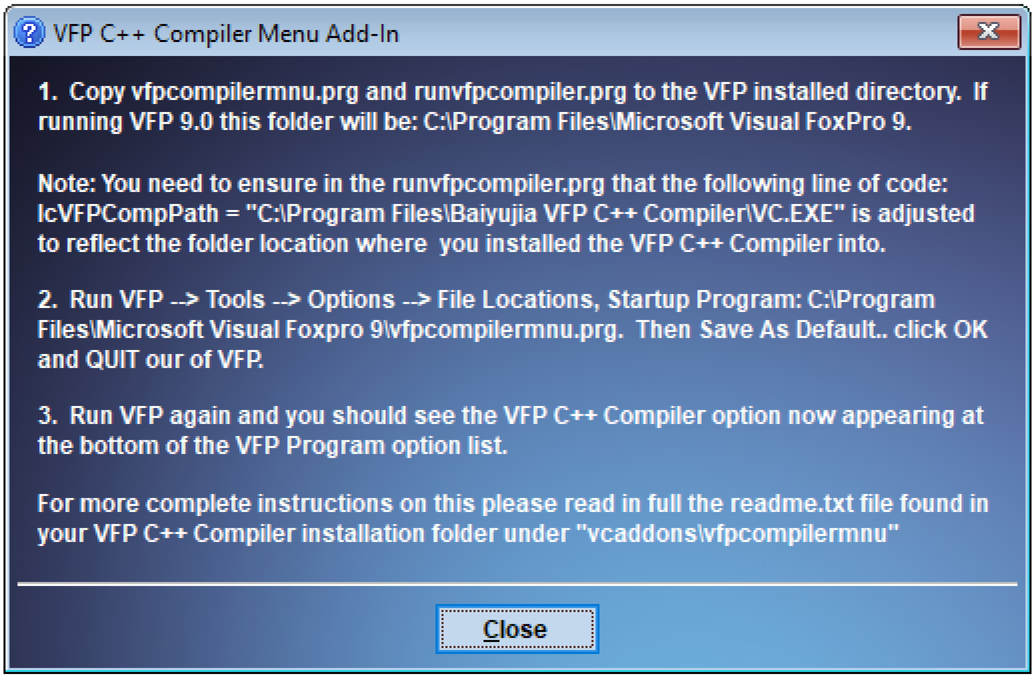
Pick the Installation Folder for VFP C++ Compiler Below Here

Do you want ther VFP C++ Compiler's Launch Option intergrated into the Visual FoxPro menu?

Click the button for the installation instuctions for this add-in:

[Installation Progress Informtion]

Figure 8: VFP C++ Installation Wizard, Options



1. Copy vfpcompilermnu.prg and runvfpcompiler.prg to the VFP installed directory. If running VFP 9.0 this folder will be: C:\Program Files\Microsoft Visual FoxPro 9.

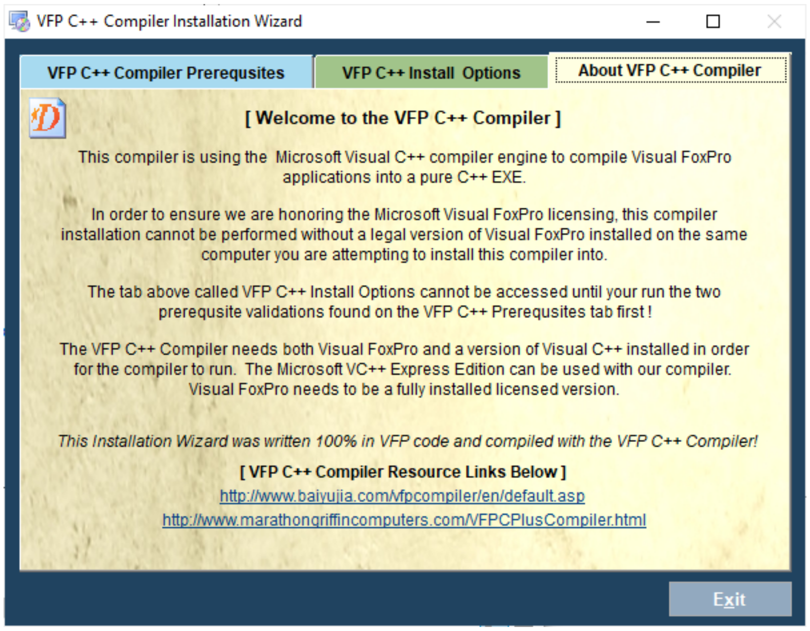
Note: You need to ensure in the runvfpcompiler.prg that the following line of code: lcVFPCompPath = "C:\Program Files\Baiyujia VFP C++ Compiler\VC.EXE" is adjusted to reflect the folder location where you installed the VFP C++ Compiler into.

2. Run VFP --> Tools --> Options --> File Locations, Startup Program: C:\Program Files\Microsoft Visual Foxpro 9\vfpcompilermnu.prg. Then Save As Default.. click OK and QUIT our of VFP.

3. Run VFP again and you should see the VFP C++ Compiler option now appearing at the bottom of the VFP Program option list.

For more complete instructions on this please read in full the readme.txt file found in your VFP C++ Compiler installation folder under "vcaddons\vfpcompilermnu"

Figure 9: Menu Add-In Instructions


VFP C++ Compiler Installation Wizard
VFP C++ Compiler Prerequsites
VFP C++ Install Options
About VFP C++ Compiler

[Welcome to the VFP C++ Compiler]

This compiler is using the Microsoft Visual C++ compiler engine to compile Visual FoxPro applications into a pure C++ EXE.

In order to ensure we are honoring the Microsoft Visual FoxPro licensing, this compiler installation cannot be performed without a legal version of Visual FoxPro installed on the same computer you are attempting to install this compiler into.

The tab above called VFP C++ Install Options cannot be accessed until your run the two prerequsite validations found on the VFP C++ Prerequsites tab first !

The VFP C++ Compiler needs both Visual FoxPro and a version of Visual C++ installed in order for the compiler to run. The Microsoft VC++ Express Edition can be used with our compiler. Visual FoxPro needs to be a fully installed licensed version.

This Installation Wizard was written 100% in VFP code and compiled with the VFP C++ Compiler!

[VFP C++ Compiler Resource Links Below]

http://www.baiyujia.com/vfpcompiler/en/default.asp

http://www.marathongriffincomputers.com/VFPCPlusCompiler.html

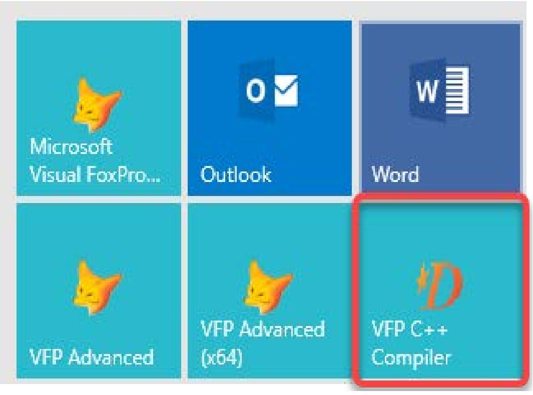
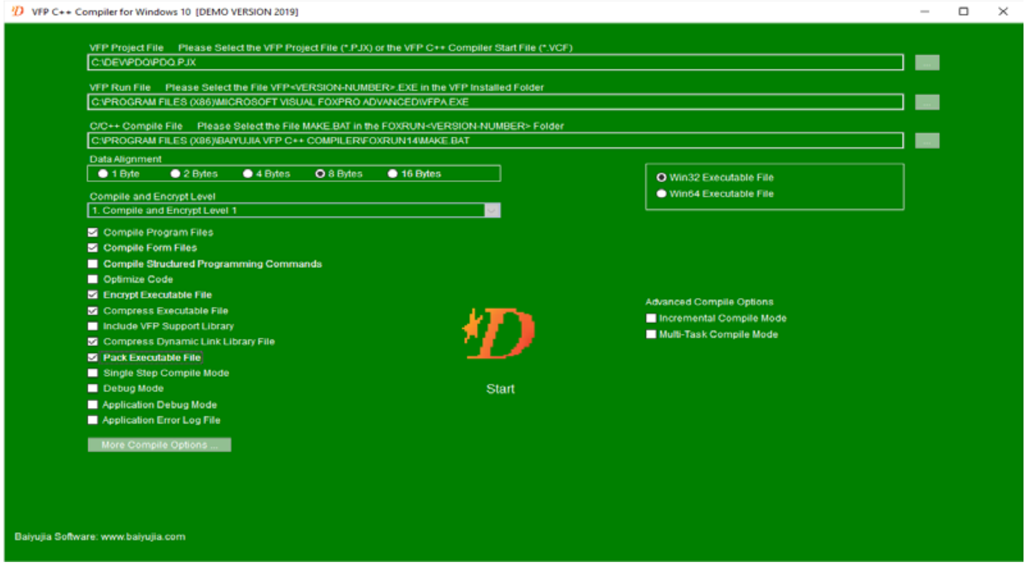


Figure 10: There's now a fancy new icon on my Start Menu to go along with my two new Visual FoxPro Advanced icons

Figure 11: More About the VFP C++ Compiler Installation Wizard



VFP C++ Compiler for Windows 10 [DEMO VERSION 2019]

VFP Project File   Please Select the VFP Project File (*.PJX) or the VFP C++ Compiler Start File (*.VCF)
C:\DEV\PDQ\PDQ.PJX

VFP Run File   Please Select the File VFP<VERSION-NUMBER>.EXE in the VFP Installed Folder
C:\PROGRAM FILES (X86)\MICROSOFT VISUAL FOXPRO ADVANCED\VFPA.EXE

C/C++ Compile File   Please Select the File MAKE.BAT in the FOXRUN<VERSION-NUMBER> Folder
C:\PROGRAM FILES (X86)\BAIYUJIA VFP C++ COMPILER\FOXRUN14\MAKE.BAT

Data Alignment
○ 1 Byte   ○ 2 Bytes   ○ 4 Bytes   ◉ 8 Bytes   ○ 16 Bytes

◉ Win32 Executable File
○ Win64 Executable File

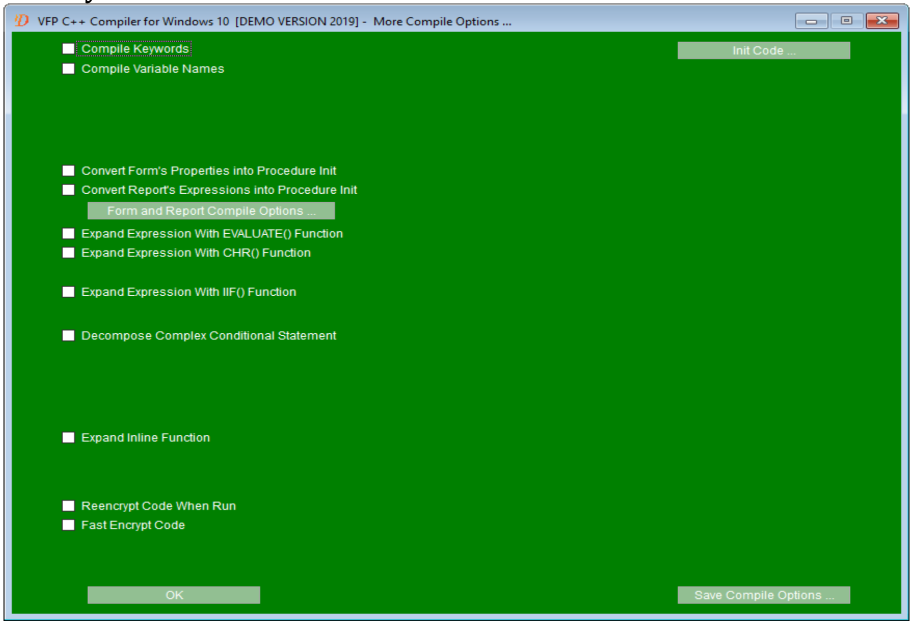
Compile and Encrypt Level
1. Compile and Encrypt Level 1

☑ Compile Program Files
☑ Compile Form Files
□ Compile Structured Programming Commands
□ Optimize Code
☑ Encrypt Executable File
☑ Compress Executable File
□ Include VFP Support Library
☑ Compress Dynamic Link Library File
☑ Pack Executable File
□ Single Step Compile Mode
□ Debug Mode
□ Application Debug Mode
□ Application Error Log File

Advanced Compile Options
□ Incremental Compile Mode
□ Multi-Task Compile Mode

Figure 12: The VFP C++ Compiler

Firing up the Compiler gives you a user interface with myriad options for compilation (**Figure 12**) and even more options (**Figure 13**). This paper is about VFP Advanced and not the VFP C++ Compiler, so I won't go through each option in detail but I wanted to at least make you aware of this tool.



VFP C++ Compiler for Windows 10 [DEMO VERSION 2019] - More Compile Options ...

□ Compile Keywords
□ Compile Variable Names
□ Convert Form's Properties into Procedure Init
□ Convert Report's Expressions into Procedure Init
□ Expand Expression With EVALUATE() Function
□ Expand Expression With CHR() Function
□ Expand Expression With IIF() Function
□ Decompose Complex Conditional Statement
□ Expand Inline Function
□ Reencrypt Code When Run
□ Fast Encrypt Code

Figure 13: Even More Compile Options!

Get a demo copy of VFP C++ Compiler at http://www.baiyujia.com/f_download.asp. If you're developing commercial software and want to ensure it runs as fast as possible with as little chance as possible of being decompiled, this will be a very useful tool and well worth the USD $499.

Bonus Bonus: VFP Encryptor

Chen also has another, free, utility on his website called VFP Encryptor. He purports that if you run your VFP executable through this utility, it will be virtually undecompilable. *[There's an interesting discussion about this at https://www.foxite.com/archives/who-can-hack-this-one-0000260349.htm*]

VFP Encryptor is a command line utility with the following options:

```
/VEENCRYPT=ON|OFF Encrypt Executable File
/VEPACKEXE=ON|OFF Compress Executable File
/SILENT=ON|OFF Silent Mode
```

Get it from http://www.baiyujia.com/f_download.asp

Appendix A: List of All Fixes in Visual FoxPro Advanced

[as of August 31, 2019]

Here is a list of the fixes. For a detailed, well-documented insight into each, go to http://baiyujia.com/vfpdocuments/ and select the fix number (See Appendix B e.g.). *Green numbers indicate new functions or commands for VFPA.*

1	Report Footer: Blank space between detail and page footer that wasn't there in VFP 8 suddenly appears in VFP 9.
2	Macro Substitution doesn't work if there's a procedure larger than 64kb in the same program.
3	A form won't run correctly if there's a procedure larger than 64kb in the form
4	In vfp9 (and vfp6, vfp7, vfp8), if the RelationalExpr expression includes macros, EVALUATE(), or TYPE() functions, VFP will crash.
5	VFP debugger stops at invisible breakpoints
6	In VFP9, the Page Setup Dialog Box has no the Printer button on Windows Vista or later
7	Varbinary or Blob Data may cause VFP to enter infinite loop if STATUS BAR=OFF and TALK=ON
8	Find or Replace Dialog Box will sometimes crash VFP if "Match Whole Word" is selected in Double-Byte Character Set environments
9	Allow REPLACE command to use up to 255 fields
10	Access and Assign methods may crash VFP if STATUS BAR=OFF and TALK=ON
11	Change SET TALK and SET SAFETY to more reasonable defaults for a private data session
12	Don't allow FOPEN() to return a file handle number of 0 (Zero)
13	Return more accurate information for OS() Function on newer Operating Systems
14	If we set the Grid.Optimize property to true (.T.), sometimes VFP will crash
15	Convert Property Names to Lowercase in the NTI to allow case-sensitive languages to use COM correctly.
16	ComboBox Control may enter an infinite loop

17	SET COLLATE option not saved to registry if set to MACHINE
18	“Shadow” of dockable windows lags far behind
19	SYS(9000) Enables or Disables shadows and docking the Dockable windows automatically
20	Always show Header.ToolTipText Property
21	SYS(9079) cleans the System Data Buffer [internal use only]
22	SET RESOURCE OFF at startup
23	SYS(9001) Enables or Disables adjusting the precision of floating point numbers
24	Precision of ROUND()
25	Main VFP form is always visible at startup if Command Window is docked and shown
26	Memo fields corrupted when part of SELECT… UNION if BLOCKSIZE > 1
27	EditBox Control / Modify Command windows may enter an Infinite Loop if non-printable characters are present
28	Change default location of FoxUser.dbf, _command.prg, and FoxTask.dbf to the user’s %APPDATA% folder rather than the startup folder
29	Expand maximum number of pictures a ListBox Control can display when RowSourceType = 5 (Array) from <500 to 65,000
30	Top Level Form (ShowWindow=2) Caption sometimes disappears in Windows 10
31	Sometimes cannot open Font Property Page for ActiveX control
32	GETPEM() cannot return the user-defined method code in a program
33	ASTACKINFO() does not return call stack information for ON events (ON ERROR, ON KEY LABEL, ON SELECTION, etc.)
34	DO FORM command from ON SELECTION BAR may crash VFP9 SP2
35	COPY FILE always creates the file with lowercase name
36	RETURN TO in Destroy event may not return to correct procedure

37	SYS(9002) Enables or Disables rounding the Datetime data (see Fix 38) to integer seconds
38	Round the Datetime seconds to integer
39	Set PageFrame.RightToLeft property to correct default (.F.)
40	“Report Objects are Too Large” error if scaling > 100% and REPORTBEHAVIOR = 90 on Windows 8.1 and 10
41	Page.Enabled Property not refreshed
42	SYS(9003) Enables or Disables fixing the datetime data (see Fix 43) for the 32-bit VFPOLEDB driver (there is no 64-bit VFPOLEDB driver)
43	Time data is lost for Datetime data in the VFPOLEDB Driver
44	SYS(9004) Enables or Disables fixing the CAST() Function for the VFPOLEDB Driver (see Fix 45)
45	CAST() function for the VFPOLEDB Driver does not convert Characters to Date
46	VARCHAR(MAX) Fields from SQL Server are Mapped as C(0) Fields
47	Selected Report Objects are printed with a shadow if REPORTBEHAVIOR = 80
48	SYS(9005) Shows or Hides the “Printing….” Window if REPORTBEHAVIOR = 80
49	Allow the “Printing…” window to be hidden (see Fix 48)
50	SYS(9006) Shows or Hides the Print Preview Toolbar
51	Allow the Print Preview Toolbar to be hidden (see Fix 50)
52	SYS(9007) Enables or Disables the Print button in the Print Preview Toolbar
53	Allow the Print button in the Print Preview Toolbar to be hidden (see Fix 52)
54	SYS(9008) Enables or Disables maximizing the Report Preview Window automatically
55	Maximize the Report Preview window automatically if enabled (see Fix 54), otherwise show it in its current size and position
56	SYS(9009) Enables or Disables fixing the “Report Objects are Too Large” error (see Fix 40)

57	SYS(9010) Enables or Disables Fixing the Cursor Position is Too Small (see Fix 58)
58	The Cursor Position is too small when scaling to 125%
59	SYS(9080) Enables or Disables releasing the COM Object in 64-bit VFPA
60	ComboBox control sometimes does not allow you to select an item
61	SYS(9011) sets or returns the maximum number of window items to display in the Window menu (also settable in Registry)
62	Allow the maximum number of Window items to display in the Window Menu to 255 (see Fix 61)
63	SYS(9012) sets or returns the direction of the List portion of the ComboBox Control either above or below the ComboBox control itself (see Fix 64)
64	Allows the List Portion of the ComboBox Control to be explicitly displayed above or below the control, rather than just letting VFP decide automatically
65	SYS(9013) Enables or Disables the FILETOSTR() function to use the Windows System Cache (see Fix 66)
66	Allow FILETOSTR() to use the Windows System Cache rather than the CreateFileA Windows API function, which may trigger anti-virus programs
67	SYS(9014) Enables or Disables the STRTOFILE() function to use the Windows System Cache (see Fix 68)
68	Allow STRTOFILE() to use the Windows System Cache rather than the CreateFileA Windows API function, which may trigger anti-virus programs
69	SYS(9015) sets or returns the number of retry attempts after DELETE FILE (or ERASE) fails, which may be caused by anti-virus programs
70	Allow variation of the number of times to try again after DELETE FILE Fails (see Fix 69)
71	SYS(9016) sets or returns the number of times to retry after MKDIR / MD command fails
72	Allow variation of the number of times you can try again after MKDIR / MD fails
73	SYS(9017) sets or returns the number of times to try again after RMDIR / RD fails
74	Allow variation of the number of times to try again after RMDIR / RD fails

75	Open File or Save File Dialog Box only shows title bar, not dialog, on some versions of Windows 10
76	SYS(9018) enables or disables the VERSION() function to return the new Version Information (see Fix 77). Disable for backward compatibility.
77	VERSION() returns the accurate Version Information for VFPA
78	SYS(9019) enables or disables the OS() function to return the accurate Operating System Information (see Fix 13). Disable for backward compatibility.
79	SYS(9020) enables or disables the ability to expand TMPFILES, EDITWORK, PROGWORK, or SORTWORK with extra environment variables unique to VFPA (VFPPROCESSID, VFPGUID, VFPTIMESTAMP, or VFPPROCNAME). This allows each process or even procedures to have its own environment settings (see Fix 80, 81)
80	Allow TMPFILES, EDITWORK, PROGWORK, and SORTWORK variables to include environment variables that are expanded when needed (see Fix 79)
81	SYS(9021) explicitly sets TMPFILES, EDITWORK, PROGWORK, or SORTWORK programmatically (see Fix 80)
82	Command Line options max length expanded to 4,095 bytes (up from 255)
83	Expand the Command Window to 10 rows, 80 columns if no resource file says otherwise (up from default of 5 rows, 50 columns)
84	SYS(9022) enables or disables the BUILD EXE command to use the Windows System Cache (see Fix 85)
85	BUILD EXE command uses the Windows System Cache rather than write to disk immediately, which slows it down and may interfere with anti-virus programs
86	Add bitwise expressions for the CALCULATE command
87	SYS(9023) enables or disables closing the file handle _TEXT before entering the ON ERROR routine (see Fix 88)
88	Allow disabling the automatic closing of the file handle _TEXT before entering the ON ERROR routine
89	SYS(9024) enables or disables reading data from an Executable file when at End-of-File (see Fix 90)
90	VFP versions 6-9 up to SP2 will try to read data from an Executable file even when at End-of-File

91	SQLSETPROP(0, “Transactions”,1) my throw error 1541 for some combinations of Windows and ODBC drivers

Appendix B: Sample Fix Report

This is Fix 6. I chose this one because it's relatively short compared to the rest, so you can imagine how much work Chen has put into these! All of these reports are available at http://baiyujia.com/vfpdocuments/.

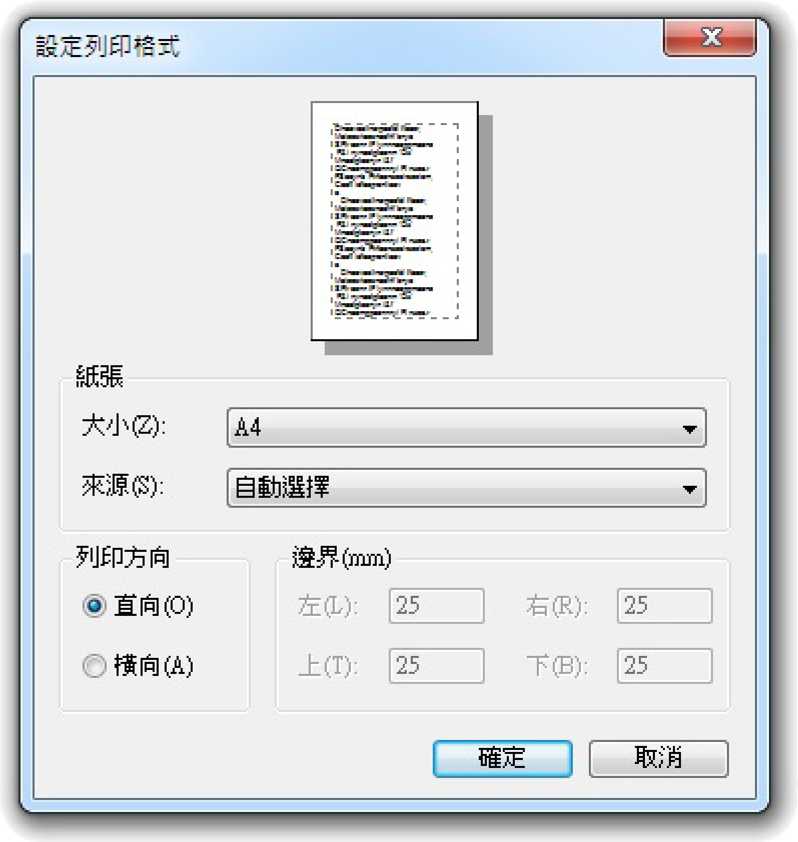
```
                 -------------------------------------------------

                       VFP 9.0 FIX - PAGE SETUP DIALOG BOX

                                  January 2019

                 -------------------------------------------------

                                      CCB
```

1. BUG:

     In vfp9, the Page Setup Dialog Box has no the Printer button on Windows Vista or later.

     There is no the bug in vfp6, vfp7 and vfp8.

     Please refer to the picture pagesetup9.jpg:



設定列印格式

紙張

大小(Z): A4

來源(S): 自動選擇

列印方向

● 直向(O)

○ 橫向(A)

邊界(mm)

左(L): 25　右(R): 25

上(T): 25　下(B): 25

2. CAUSE:

In vfp6, vfp7 and vfp8, it call the PrintDlgA Windows API to show the Page Setup Dialog Box.

In vfp9, it call the PageSetupDlgA Windows API to show the Page Setup Dialog Box.

But for some about security reasons, the Printer button has been removed on Windows Vista or later.

3. RESOLUTION:

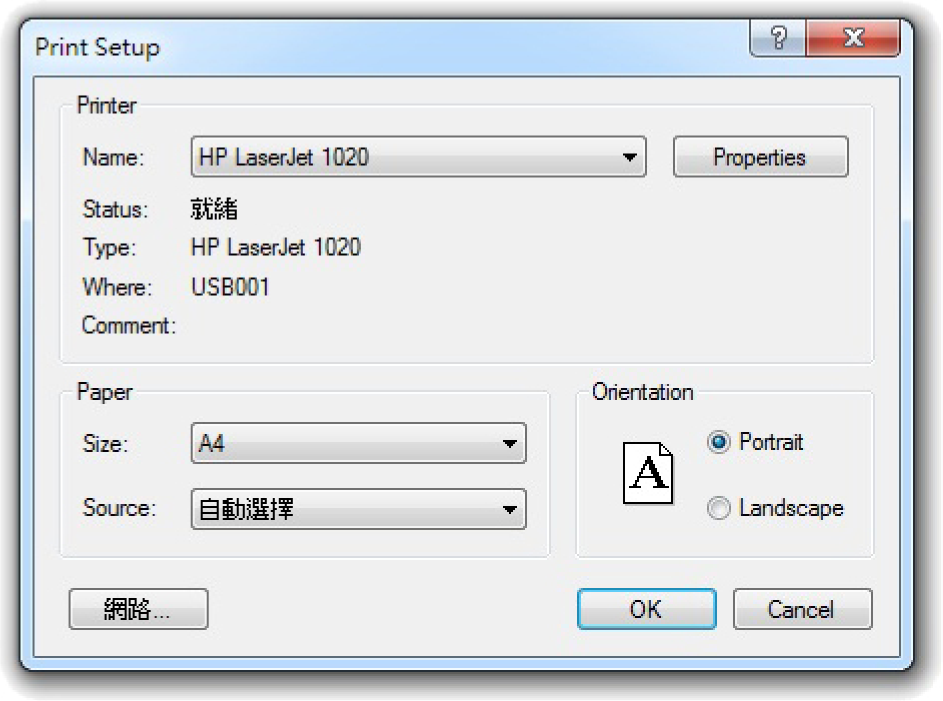
We can write some code to fix the BUG.

```
vfpa_pagesetupdlga proc
push ebp
mov ebp,esp
sub esp,48h
push eax
push ebx
push ecx
push edx
push esi
push edi
pushfd

mov esi,dword ptr [ebp+08h]
lea edi,dword ptr [ebp-48h]
xor eax,eax
cld
movsd
sub dword ptr [edi-4],(54h-42h)

movsd
movsd
movsd

stosd

movsd
; PD_PRINTSETUP (0x00000040) + PD_ENABLESETUPTEMPLATE (0x00008000)
mov dword ptr [edi-4],00000040h+00008000h

stosw
stosw
stosw
stosw
stosw
add esi,4*2+4*4+4*4

movsd
mov dword ptr [edi-4],00400000h
movsd

stosd
movsd
add esi,4

stosd
movsd
mov dword ptr [edi-4],0616h
```

```
stosd
movsd

lea eax,dword ptr [ebp-48h]
push eax
call PrintDlgA

mov dword ptr [ebp-04h],eax

lea esi,dword ptr [ebp-48h]
mov edi,dword ptr [ebp+08h]
xor eax,eax
cld
movsd
add dword ptr [edi-4],(54h-42h)

movsd
movsd
movsd

add esi,4

movsd
; PSD_DISABLEMARGINS (0x00000010)
mov dword ptr [edi-4],00000010h

add esi,2*5
add edi,4*2+4*4+4*4

movsd
movsd

add esi,4
movsd
add edi,4

add esi,4
movsd

add esi,4
movsd

popfd
pop edi
pop esi
pop edx
pop ecx
pop ebx
pop eax
mov eax,dword ptr [ebp-04h]
mov esp,ebp
pop ebp
ret 04h

vfpa_pagesetupdlga endp
```

We can use the vfpa_pagesetupdlga() function instead of the PageSetupDlgA Windows API, then the Page Setup Dialog Box will show the Printer and Properties buttons.

Please refer to the picture pagesetup.jpg:



Print Setup

Printer

Name: HP LaserJet 1020

Status: 就緒

Type: HP LaserJet 1020

Where: USB001

Comment:

Paper

Size: A4

Source: 自動選擇

Orientation

● Portrait

○ Landscape

4. APPLIES TO:

VFP 9.0.0.2412

VFP 9.0.0.3504 (SP1)

VFP 9.0.0.4611 (SP2)

VFP 9.0.0.5015 (SP2)

VFP 9.0.0.5411 (SP2)

VFP 9.0.0.5721 (SP2)

VFP 9.0.0.5815 (SP2)

VFP 9.0.0.6303 (SP2)

VFP 9.0.0.6602 (SP2)

VFP 9.0.0.7423 (SP2)

The bug has been fixed in VFP Advanced.

5. REFERENCE WEBSITES:

1, baiyujia.com:

http://www.baiyujia.com

2, foxite.com:

http://www.foxite.com/archives/sys1037-not-having-printers-button-0000158140.htm

http://www.foxite.com/archives/printer-button-on-page-set-up-sys1037-0000318953.htm

3, microsoft.com:

http://connect.microsoft.com/VisualStudio/feedback/details/286313/vfp-sys-1037-no-longer-has-a-printer-button-in-vista

http://social.msdn.microsoft.com/Forums/en-US/adcf515c-a1f3-4dce-a2b6-b0ad8286fb42/cant-switch-printers-from-vfp

6. OTHER:

For reference only, there is no guarantees.

Any questions or suggestions, please send me an email at ccb2000@163.com.