

Adding Voice Response Capabilities to Your FoxPro Applications

By Whil Hentzen

Your initial reaction when hearing an automated voice as a result of a phone call is usually one of disgust ("I just want to talk to Herman, I don't want to go through voice mail hell...") or relief ("I just want to find out my balance - I don't want to wait for five minutes while some idiot on minimum wage interrogates me about account numbers and last deposit made and what my mother's maiden name is.")

The mechanism that produces the second reaction - an automated telephone answering and querying system - is known as an interactive voice response (IVR) application, and until recently, was the domain of large companies with large budgets to spend expensive, customized systems. Voysys Corporation's voysAccess 1.0 for FoxPro is going to change that.

For just \$999, a developer can buy a two-line telephone board and the software libraries required to build robust IVR applications for the desktop. I first saw a demonstration of voysAccess in February, and was immediately struck by the ease of use and flexibility that voysAccess provided.

voysAccess consists of two components: a full-length voice board manufactured by Dialogic that contains jacks for two incoming telephone lines, and a FoxPro/Windows 2.6 .FLL library that provides access to telephony functions from within FoxPro. After installing the board, you simply use the SET LIBRARY TO command inside your app and you've got the ability to make calls to eight voysAccess functions. These functions allow your FoxPro application to communicate with a phone call through the Dialogic board connection. These functions include answering the phone, looking for a hang-up, playing a voice message, and storing the values that the user has entered using their touch-tone phone. The values can then be used to search for, manipulate and return data from the application.

Hey, I know you - you don't want to hear a lot of marketing talk and high-falutin' promises - let's get to the code to see how this is actually implemented. The following code (simplified to eliminate some error checking and initializations) is all that's need to answer the phone, ask the caller for an account number, give the caller three chances to enter it successfully, and then verify that the number is valid.

```
* sample.prg
* the "V" functions are all part of the voysAccess library
*
* open the account table and the voysAccess library
*
use ACCOUNTS order cNoAcct
set library to \voysaccs\core\voysaccf.fll

*
* sit by the phone, waiting for it to ring...
*
do while m.nRetVal < 0
  m.nRetVal = VWaitRing(1, 10)
enddo

*
* say hello
*
m.nRetVal = VSpeak("ptHello")
```

```

*
* ask for a 4-digit worker ID until we get a good one
* the caller has three tries before we hang up
*
for m.nCount = 1 to 3

  if vChkHangup() = 1
    return
  endif

  *
  * get a 9-digit account number from the caller
  * with an inter-digit timeout of 15 seconds
  * in case the caller decides to just sit there
  *
  m.nRetVal = VSpeak("ptNoAcctGet")
  if VGetTones(@cNoAcct, 9, " ", 15) < 0
    return
  endif

  *
  * verify that the account number entered is in the table
  *
  if seek( m.cNoAcct )
    exit
  endif

  *
  * if not found, tell the caller and let them try two
  * more times (on the third try, hang up)
  *
  if m.nCount < 3
    m.nRetVal = VSpeak("ptNoAcctBad")
  else
    *
    * the caller couldn't get it right, so bail from
    * this sample program
    *
    return
  endif

endfor

.
.
.
< additional functions now that we've got a good account number >
.
.
.
*
* If the caller is still on the line, say "goodbye"prompt.
*
if VChkHangup() = 0
  m.nRetVal = VSpeak("ptbye")
endif

```

```
*  
* hang up the phone  
*  
m.nRetVal = VHangup()
```

It really is this easy - I wrote this program in about 15 minutes, using the sample application as a guide.

The one unfamiliar piece in the above code is the parameter used in several of the voysAccess function calls. Each of these parameters represents a prerecorded prompt, such as "Please enter your Account Number." voysAccess comes with over 160 prompts and a tool, voysSmith, that lets you create your own if the defaults aren't enough.

IVR applications are fairly straightforward - the functions used above are perhaps all you'd need in a sophisticated application. The additional complexity comes from developing the logic that the caller will encounter - "What if they get all the way to the end, but then want to abort all transactions except the first?" - and dealing with inevitable hardware problems such as noisy telephone connections, power losses and networks crashing.

Installation of voysAccess, because it required opening up the computer, was more of a pain than I'd like. It took about an hour to install the board, get it configured so that the interrupts and memory offsets were set properly, and get it to answer a call, but remember, I'm a software developer because I don't like hardware. Once the board was in and the telephone line was connected, it took just a few minutes to install the software and successfully run each of the demo programs.

The documentation was basically clear and straight forward, although the version I tested had four pamphlets for different aspects of the installation. Once up and running, I hardly used the manuals. If I was creating a full-featured system, I'd have dog-eared the "Building Successful IVR Applications" text that explained how to find, justify, and sell IVR applications as well as how to design them properly. This manual even included a section on identifying applications that are not candidates for IVR. A lot of products have a technical reference, but it's the rare tool that also explains how to use it strategically.

I really like this product. It's easy to use and the applications are fun and innovative to write. The only problems are more environmental than FoxPro-related. For example, you need between 64 and 128K to load the Dialogic driver into memory, depending on your requirements, and that may cause conflicts with other memory-hogging programs in some installations. And since this is designed to work with applications that are available "round-the-clock", you'll need to take care of those situations where the computer could go down.

While currently available only for FoxPro/Windows 2.6, Voysys Corporation has announced a VBX that will allow Visual Basic developers to use voysAccess, and both a library and an OLE control for Visual FoxPro will be available this fall.

voysAccess 1.0 for FoxPro
\$999 (includes 2 line Dialogic voice board)

Voysys Corporation
48634 Milmont Drive
Fremont CA 94538

Voice 510.252.1100
Fax 510.252.1101

Tech Support 800.786.9797 x357

<EOF>