Getting Started

Building computer telephony applications with VBVoice is a simple process. Working within the Visual Basic design environment, VBVoice provides a set of components that you use to build applications by creating a visual call flow diagram. To illustrate this visual design approach and to get you started as quickly as possible, this document presents a step-by-step example of how to create a simple voice messaging application using VBVoice. This document presents the minimum amount of information in order to get you going as quickly as possible.

Overview of the VBVoice Design Approach

Designing a VBVoice application involves 4 steps:

1. Planning the system
2. Creating the Call Flow Diagram and Setting Properties
3. Creating and configuring greetings
4. Testing the system

1. Planning the System

Plan the call flow for your system and define the voice menus to be presented to the caller. You need to provide a logical structure of menus to the caller, and allow the caller the choice of correcting mistakes by re-entering digits or returning to a previous menu. You should limit the number of options presented to the caller to 5 or 6 on each menu, . Too many options create long greetings, and the caller will likely forget which number goes with each option. Once you have the call flow mapped out, you can start building your application with VBVoice.

2. Creating the Call Flow Diagram and Setting Properties

Building a VBVoice applications begins with the creation of a Visual Basic project (.VBP extension). Visual Basic projects use one or more forms (.FRM file extension), onto which system functionality and the VBVoice call flow diagrams are defined. The Visual Basic toolbox contains coded objects called controls. Controls contain properties that are set by the designer to define the environment of the program. Examples of controls include text boxes, scroll bars and push buttons. VBVoice adds its own set of powerful computer telephony controls to the Visual Basic toolkit.

The Call Flow Diagram for your application is built by placing VBVoice controls onto a form, configuring the controls, and then graphically connecting them according to the design plan. Every VBVoice control performs a specific task related to the voice system. When a control has performed its task, it passes the call flow to the next control in the call flow diagram.

Each VBVoice control has characteristics, or properties, that must be set to determine the behavior of the controls at run time. Each VBVoice control used in an application must be given a unique name in order to identify it and make its properties accessible. VBVoice provides a powerful mechanism called control property substitution that allows VBVoice controls to reference and access the properties of other VBVoice controls. At design time, VBVoice control properties are set via easy to understand setup dialogs.

3. Creating and Configuring Greetings

The eventuality of voice interaction with the user requires that greetings be planned and recorded. VBVoice controls that interact directly with a caller usually have an initial greeting (the EntryGreeting) that is played when the call is transferred to that control. A VBVoice greeting is made up of one or more voice phrases that are played in sequence. The play usually terminates when the caller presses a key. Some controls have other greetings that are played to prompt the caller for more input, or to inform of errors, etc.

4. Testing the System

Using VBVoice Test Mode, you can test your design at any step of the development process and from within any control. A sound card test mode provides an on-screen dial pad to allow you to simulate caller input while it plays the greetings over your sound card.

The E-Z-Flow Software Company Example

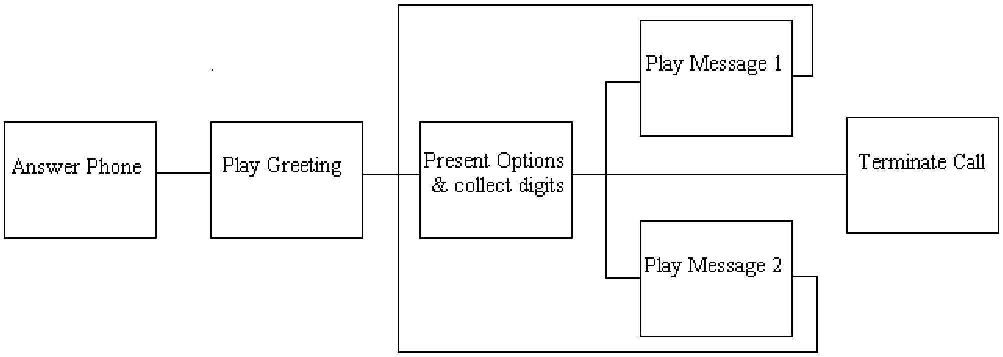
To help introduce the VBVoice interface, let's walk through a simple application. We recommend you try building the project as we present it.

1. Planning the System

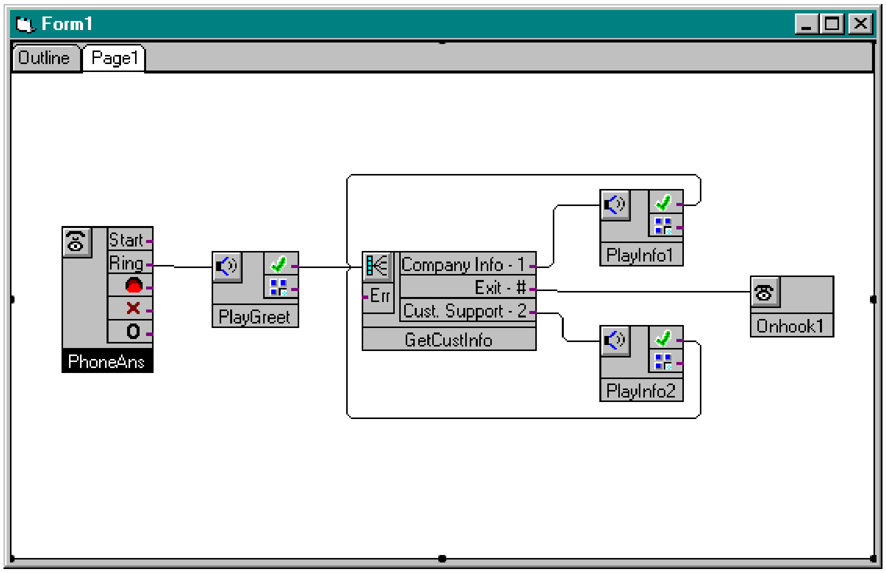
Case: The E-Z-Flow Software Company

The president of E-Z-Flow Software Company is looking to improve after hours service available to current and prospective clients. After meeting with the system developer, the decision is made to create an audiotext system that will use a voice menu to guide callers through a selection of company, product and support information. The system will prompt the caller to enter a selection and will respond to the caller's choice by playing the appropriate message. The caller will then be returned to the original menu and be re-prompted until the option to terminate the call is selected.

The key processes in this system are: answering the phone, playing a greeting, prompting for an option, collecting digits, processing the selection (playing the appropriate message) and terminating the call. The call flow may be mapped out like this:



Let's build the application and see how the award-winning VBVoice user interface turns a simple flowchart into the fully operational telephony application shown below.



2. Creating the Call Flow Diagram and Setting Properties

Creating a VBVoice project

To create a new project in VBVoice, select New Project from the File menu in Visual Basic. This opens a new project with default Visual basic properties. Now from the Tools menu, choose Custom Controls. PronexusVBVoice should already be in the list of available controls as a result of the installation process. If it is not, it can be added by choosing Browse and selecting VBV32.OCX. Select PronexusVBVoice so that it has it's check box checked. Press OK to load the VBV32 controls to the Visual Basic toolbox.

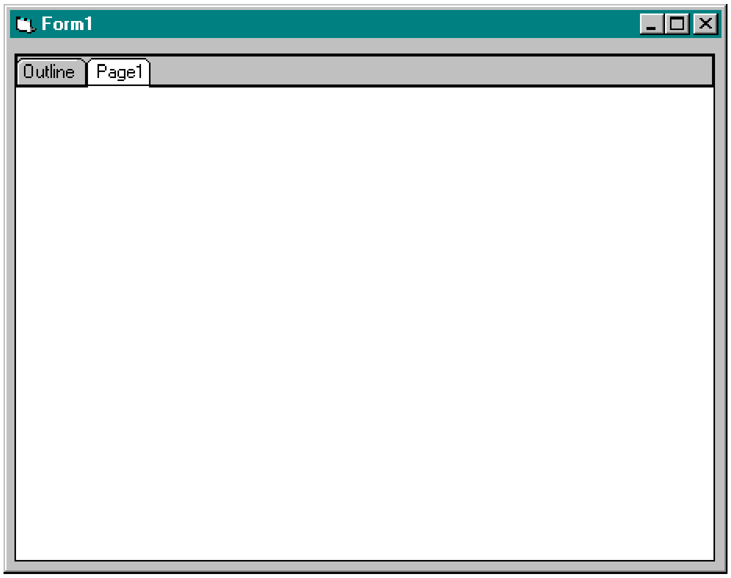
To have VBV32.OCX load by default onto new projects, open the AUTOLOAD.VBP project in your VB directory, select the controls that you wish to load as default, and then save the project.

When opening a new project, Visual Basic generates Form1, which should be visible on the screen. Save the form file now by selecting the Save File As option from the File menu and renaming it c:\vbv32ev\ezflow1.frm. Save the project now by selecting Save Project As from the File menu and renaming it c:\vbv32ev\ezflow1.vbp.

Adding the VBVFrame control

This control is a container for all other VBVoice controls. The VBVFrame creates one design page by default onto which controls are added and connected to create the call flow diagram. The frame also presents an outline view of the system and the capability of adding more pages.

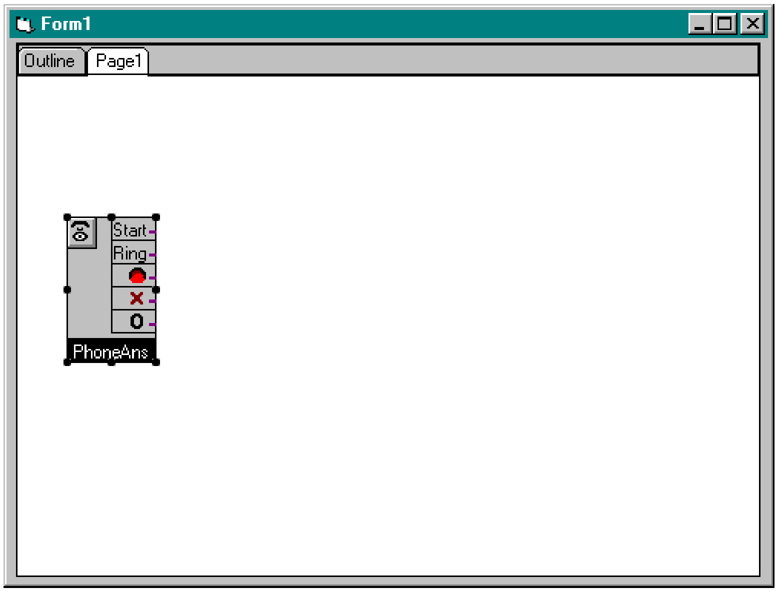
Draw a VBVFrame on the form by selecting (clicking once on and releasing) the VBV icon and then placing your cursor on the form and dragging your mouse across the form with the left mouse button depressed.



Adding VBVoice controls onto the frame

There are 6 VBVoice controls active in this application. To place a control on the page, select (click once on and release) the desired control and then position your cursor on the form and drag your mouse (left mouse button depressed) to draw a box on the page.
Note: Do not double click on the toolbar button: this will add the control directly to the form, not the frame.
Note: If your system does not display the control name, change the environment setting now, by right-clicking on a blank area on the page, selecting Properties and checking the Show Control Names box. If your system does not play sound properly set your **Default Voice File format** to 8K uLAW.

We begin by adding a **LineGroup** control to pick up the phone when it rings (incoming calls are automatically routed to the 'Ring' input node of this control). Select a LineGroup control from the toolbox now and place it onto the form.



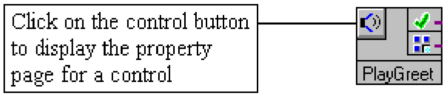
Setting control names and properties

Each VBVoice control must be given a unique name in order to uniquely identify the control and its properties. Some of the properties are used to configure the control at design time, and others are used to change the behavior of the system at runtime by Visual Basic code. (A detailed discussion about Controls and their properties is presented in subsequent sections of the manual). Control names are accessed using the control context menu. Control properties are accessed through the context menu or by clicking on the icon button in the upper left corner of the control.

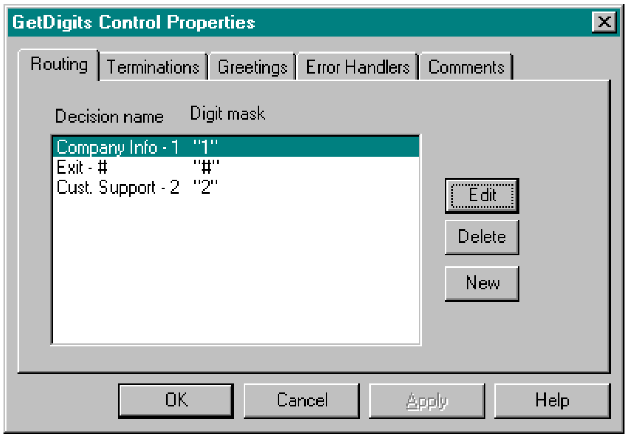
Activate the context menu now by right clicking on the LineGroup control that you placed on the page. Change the control name by selecting Rename Control from the menu, and typing "PhoneAns" at the prompt. Press OK.

A **PlayGreeting** control will greet the caller with a short message and pass call flow to the next control. Select a **PlayGreeting** control from the toolbox now and place it onto the form. Change the name of the control to 'PlayGreet'. Press OK.

Next, you will add a **GetDigits** control to present a menu of options to the caller, including company and product information, customer support numbers to try and an exiting option. The control then accepts and processes a selection from the caller. The options are established in the Setup dialog associated with this control.


Click on the control button to display the property page for a control

Select a GetDigits control from the toolbox and position it on the page. Change the name of the control to GetCustInfo. Click on the control button and select the Routing property to create the options for this control.



Decision name	Digit mask
Company Info - 1	"1"
Exit - #	"#"
Cust. Support - 2	"2"

We want to limit the caller to entering one of '1', '#' or '2'. To accomplish this, you will enter a setup entry for each digit. In the Routing dialog, delete the current default entries and add a New entry. Set the digit mask, used to validate entry, to the value '1'. Set the condition name to 'Company Info - 1'. Enter OK. Repeat this process for the combinations ['#','Exit - #"] and ['2','Cust Support - 2']. Your 3 conditions should be visible in the dialog. Next, select the terminations dialog and set the Number of retries on error option to 2.

In the event the caller enters an invalid entry or does not respond to the request to select an option, we want the call routed back to the input node of the control to play the greeting again. The default processing, as determined by the Error Handlers property of this control, does exactly this, and therefore does not need to be changed.

Now you can position the 2 remaining Greeting controls and the OnHook control, onto the page.

The first **PlayGreeting** control plays a pre-recorded message containing information about the company and its products and new product releases. Rename this control "PlayInfo1".

The second **PlayGreeting** control plays a pre-recorded message providing an off-hour customer support phone number. Rename this control "PlayInfo2".

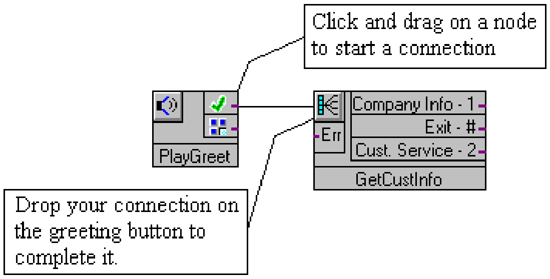
The **OnHook** Control terminates the session by placing the phone on-hook.

Connecting the controls

We are now ready to connect the controls and create the call flow. VBVoice controls have up to two input nodes on the left hand side, and one or more output nodes on the right hand side. A call enters the control via an input node, and when a control has completed processing directs the call flow through the output node to the next control. Nodes are graphically connected on the screen to indicate the call flow either by lines or via named connections (the developer uses a named connection when a graphical connection is not suitable - for instance, when the connection is an error path outside of the main call flow, or when the call moves to a module on a separate page or form).

To connect a VBVoice control to another, click on an output node and drag to the input node of the control that will perform the next voice action. VBVoice will draw a blue 'rubber-band' as you drag, and when you are positioned over an input node, will identify the connect with a small target. Release and the connection becomes permanent. Once permanent, connections to and from a node are highlighted when the mouse passes over it.

Note: Releasing the cursor on the page instead of over an input node will generate a dialog prompting you to connect to another module or form. The ability to create a named connection to another module is a standard feature of VBVoice. It is not required in this sample however, and should cancelled if you encounter it.



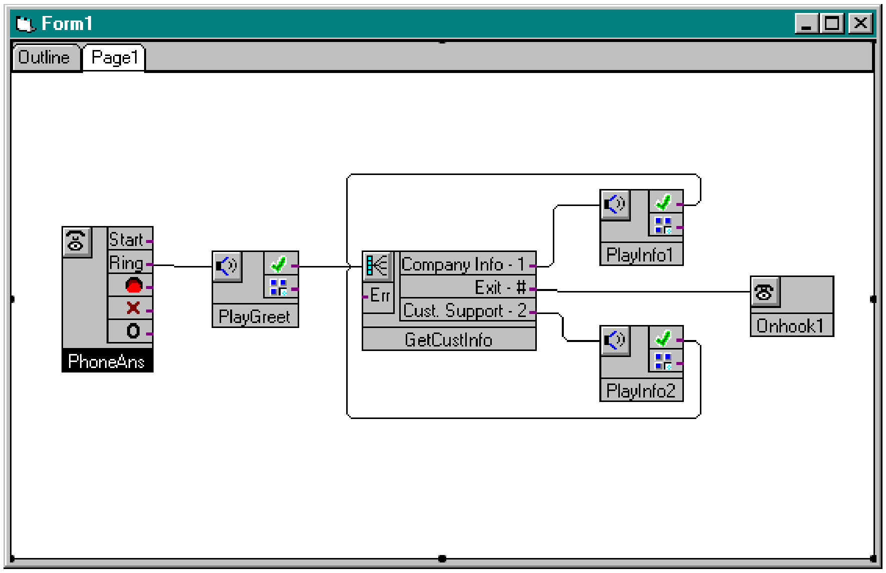
Click and drag on a node to start a connection

Drop your connection on the greeting button to complete it.

You can now connect the nodes on the E-Z-Flow example. Using the process described above, connect the following nodes now:

the Ring output node of PhoneAns to the input node of PlayGreet,
the <Check> node of PlayGreet to the Control node of GetCustInfo,
the Company Info node of GetCustInfo to input node of PlayInfo1,
the Exit node of GetCustInfo to the input node of OnHook,
the Cust Service node to the input node of PlayInfo2,
the <Check> node of PlayInfo1 to the Control node of GetCustInfo, and
the <Check> node of PlayInfo2 to the Control node of GetCustInfo

Your call flow diagram is now complete, except for the specification of greetings and testing. It should look like this:



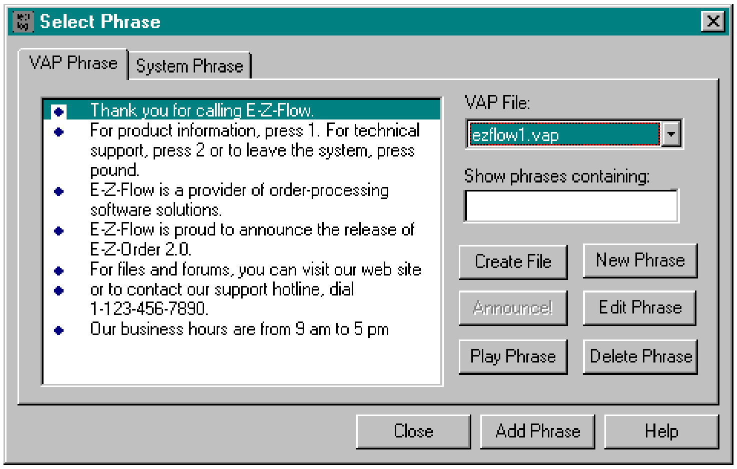
3. Creating and Configuring Greetings

It is recommended that developers record greetings in a session and then call the pre-recorded message from a file list when building the application. The Greetings property allows you to create greetings and to select from pre-recorded voice files.

Click on the control button of the PlayGreet control to display the properties dialog.

Note: For the purposes of this application, we will load Greetings from a pre-recorded indexed file of greetings. We will also add an optional phrase for those of you equipped to record dynamically. A full discussion on creating Greetings is presented in the Greetings section of the manual.

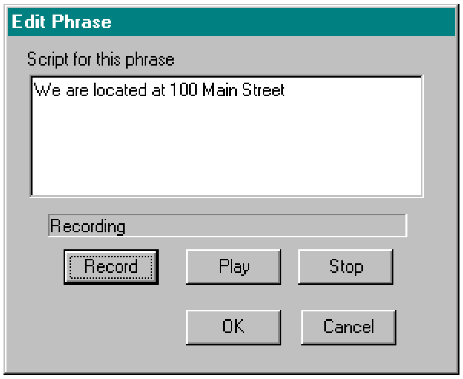
To add a pre-recorded greeting to the PlayGreeting control, select the Greetings dialog and add a phrase to the control by clicking on the Add Phrase option. This will bring up the Select Phrase dialog. Choose the ezflow1.VAP file from the list of voice files. From the phrases that appear in the window, select (double click) the one that begins "Thank you for calling…". Scroll down the list and select a second phrase, which begins "Our business hours…".



VAP File: ezflow1.vap

- Thank you for calling E-Z-Flow.
- For product information, press 1. For technical support, press 2 or to leave the system, press pound.
- E-Z-Flow is a provider of order-processing software solutions.
- E-Z-Flow is proud to announce the release of E-Z-Order 2.0.
- For files and forums, you can visit our web site
- or to contact our support hotline, dial 1-123-456-7890.
- Our business hours are from 9 am to 5 pm

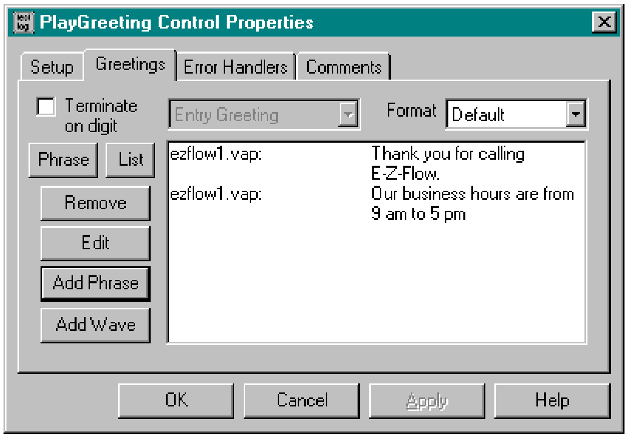
If you have the capability of recording (a sound card hooked up to a mic or a voice card hooked up to a telephone) you can add a phrase to the ezflow1.vap file now and include it into the customer greeting. Select Add Phrase from the Greetings dialog.



Script for this phrase: We are located at 100 Main Street

Type "We are located at 100 Main Street" in the script window. When you are ready to record, press the record button and speak the phrase into your recording device. Press Stop when you have finished the phrase. Press OK and select your greeting in the Select Phrase dialog.

Select Close to get out of this Select Phrase dialog. The phrases you have selected should appear in the phrase window of the greetings dialog (if you recorded the phrase above, it will also appear).



PlayGreeting Control Properties

□ Terminate on digit

Entry Greeting

Format: Default

ezflow1.vap:	Thank you for calling E-Z-Flow.
ezflow1.vap:	Our business hours are from 9 am to 5 pm

From the Control Properties dialog, select OK to terminate the phrase assignment session for this control. On returning to the main form, hold the mouse over the PlayGreet control button for a second or two to display the greeting.

Following the procedure you used in the PlayGreet control, add the entry greeting to the GetCustInfo control that begins with "For product information…" from the ezflow1.vap file.

To the PlayInfo1 control, add the greetings that begin "EZFlow is a provider of…" and "EZFlow is proud to announce…" from ezflow1.vap.

To the PlayInfo2 control, add the greetings that begin "For files and forums…" and "or to contact the support…" from ezflow1.vap.

The system is now complete and ready for testing.

4. Testing the System

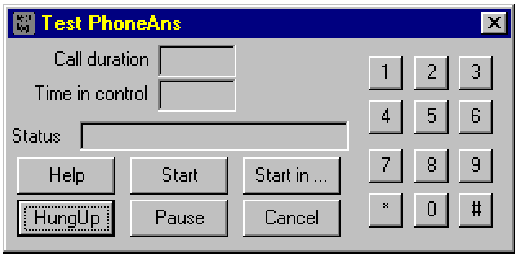
Configuring Test Mode

For testing, we will activate the VBVoice sound card test mode. To setup the test mode, access the VBVFrame context menu by right-clicking on a blank portion of the VBVFrame page. Select Properties…Test Mode to access the test mode setup sheet. Select "Use Soundcard". Switch to the "General" setup sheet of the same properties dialog and set the "Default voice file format" to 8kHz uLaw. If you do not have a sound card, it is possible to use your built-in PC speaker with a speaker driver that is available from your dealer.

Starting a Test

The Test mode allows you to start a test of your system at any point in the call flow diagram. When using the sound card, an on-screen dial pad or the keyboard can be used to enter digits.

Start the test by right-clicking over the PhoneAns control to get the context menu and select "Start Test". The Test Mode dialog will appear as shown.



Test PhoneAns

Select the Start command to begin the test. *You will notice a test log window pop up. This window monitors events as they occur in the system. If an error occurs, it will be logged in this window, and the test will stop*.

The Start Test pop-up window will prompt you to choose between Ring or Start Call. Choose Ring. Immediately, VBVoice follows the progression of the call on the form by highlighting connections as the call moves through each control.

As the test proceeds, the greetings will be played over your sound card and you can simulate caller input using the on-screen dial pad. This test should have proceeded error free. If so, congratulations...you are officially a VBVoice developer. If not, check the test log window and the call flow diagram for clues about what went wrong and revisit the application steps to rebuild the control.

To leave Test Mode and return to design mode, select Cancel from the Test Mode dialog.