Index

 Count Maintains an internal counter, which can be reset or incremented. The call is transferred to a new control depending on the results of a comparison between the counter value and a preset limit. Useful for simple loops.

 DataChange Changes data in a database record, in one or more fields. The database and record are selected by a previous DataFind or DataNew control.

 DataFind Searches for the first or the next database field depending on whether the control is entered via the main input or the "Next" input. A wide range of data match conditions is available.

 DataGet Reads a database record from a previous DataFind control, stores the result in the property 'Result', and routes according to the value obtained.

 DataNew Adds a new, blank record to a database.

 Delay Implements a wait period. The caller can be put on hold, or a short voice clip can be played during the wait.

 Dial Dials some digits either during a call or to start a call. Can perform call supervision (answer, no answer, busy etc.) and has built-in support for PBX call transfer.

 GetDigits Plays a greeting, and waits for digits from the caller. Configurable exit conditions using wild characters, numeric characters and ranges. Built-in invalid digit and no digit handling.

 Greeting Plays a greeting and exits.

 InConn Used for graphical connection of one control to another on a different form. Only InConn and OutConn controls can be connected across forms.

 IniSw Searches for a setting in a Windows initialization file. Branches to another control depending on the value found.

 Onhook . Plays a greeting, and hangs up the phone.

 OutConn Used for graphical connection of one control to another on a different form. Only InConn and OutConn controls can be connected across forms.

 Phone This control will wait for incoming ringing on a channel, or it can start a call automatically or under control of your code. After a preset number of rings, the channel is taken off hook and control is passed to the next control.

 PlayMsgs A high level control that accesses a database table containing a list of messages. Configurable options after play. Supports new, old and deleted messages. Updates database with new message status.

 Record Plays a greeting, records a file, and adds a new record to a database. At the end of the recording, allows the caller to choose options to re-record, delete, append to end of recording, or to

save the message.

 TimeSw Transfers the call to another control according to time of day and day of week.

 User This control allows you to use VB code to create your own control. You have access to all the high-level VBVoice functions and the low-level hardware-specific driver functions.

 SendFax This control is part of the VBFax option. Although included in the VBVOICE.VBX, it is inactive unless the VBFax option is purchased.

Properties List

Events

- Condition
- Decision
- DelayStarted
- Disconnect
- Enter
- EnterB
- Exit
- PhraseError
- PlayRequest
- Ring
- VoiceError

Count Properties

Counter

GotoControl

GotoNode

MaxCount Integer

Count Events

- Enter
- Exit
- VoiceError

Count control

Properties Events
Default Property: Count

Description:

This control maintains an internal counter which can be reset and incremented by call flow. It is useful for simple loops.
When a call enters the Count control via the top input, an internal counter will be reset to 0 and the call is passed directly to the control connected to the top output node (node 0). When a call enters via the second input B (+1), the Counter value is tested against the limit value. If Counter equals the limit, the call is passed to node 2 (Limit), otherwise the Counter is incremented and the call is routed to node 1 (Inc).
For example a limit of 5 means that on the fifth pass through the control, the call will be routed to the "Limit" node. The 'Limit' node is treated in the same way as the 'Invalid' node in the GetDigits control, so if the node is not connected, the invalid digit termination procedure is invoked. This consists of playing the default invalid digit greeting, and going on-hook on the line.

Properties

- GotoControl
- GotoNode

Events

- Enter
- Exit
- VoiceError

DataChange control

Properties Events
DataChange dialog

Default Property: None

Description:

Changes one or more fields in a database record. The database and record to be changed are selected by a previous DataFind or DataNew control.
If a DataChange is connected to a DataNew control, the DataNew control will create a new record, and the fields in the new record will be set to the values specified in the DataChange control.
If a DataChange is connected to a DataFind , an existing record in the database will be selected and its field values changed to the new values.
The fields to be changed, and the data to be inserted can both contain control names. If using control names within field names, ensure that valid field names will always be created. If an invalid field name is created, a VoiceError event will be generated in the control. Field names can be checked against another database table using a DataFind/DataGet combination, an IniSw control, or by VB code in the Enter event.

DataChange Dialog

See Also DataChange

Set data in ... field

This field shows the DataNew or DataFind control that will be used to locate the record to be changed.

New Fields

This box contains a list of field names in the selected database, and the new values that will be assigned to them. Use the Edit, Delete and New buttons to modify this list.

Edit command

Use the Edit command, or double click on an entry to show the Edit Conditions dialog (see page ...).

Delete command

Deletes an entry

New command

Use this command to create a new condition.

Properties

- BookMark
- DataSource
- GotoControl
- GotoNode
- RecordCount

DataFind control

Properties Events
Default Property: Count

See Also DataFind dialog

Description:

This control can select a record from a database for input validation and for subsequent data retrieval and update operations by DataNew and DataChange controls. The search can start from the current position in the database, or from the beginning. DataFind can perform searching using data collected by other controls, and can search using several different techniques. See also Using Databases .

Start at...
If the control is entered via the top input, the DataFind control will search for the first database record that matches the search criteria in the recordset of the specified data control. If the control is entered via the Next input, the control searches for the next record, starting from the previous record found. If no record had been previously found, then the database is searched from the beginning.

Database selection
The data control selected by the DataSource property defines the database file and the table to be searched using the DatabaseName and RecordSource properties in the data control.

Data search specification
If *All records* is checked, the DataFind control will scan through all records one at a time.
If *Records Containing* is checked, the DataFind control will search a field in the database for the data given, using the Match specifications provided
Data can be matched either using a digit to alphanumeric translation, (i.e. using the letters on the telephone keypad) or exact matching. Data can also be matched completely or using a 'findfirst' method, where the database field is allowed to be longer that the supplied value. Using this method, the supplied digits 12 will match database records containing 123, 1245 and 12, but not 1 or 13. Using alphanumeric translation, the data field 234 will match ADH and BEG.

Exit to...

If the record or entry is not found, the call is routed to the "Not Found" node.
If the record is found, the call is routed to the "Found" node.

Performing directory lookup
To use the DataFind control for directory lookup, use *Find record using...*
Alpha digits to perform numeric to alphabetic translation, and use *Match On: First* so it finds all the records that start with the digits that the caller enters.

BookMark

Array String(1, MAXCHANNELS)
Applies to DataFind dialog

The BookMark property is used to maintain the position of the last record found in the database. It can be set to an empty string to restart a search at the beginning, or it can be set to a valid bookmark in the data control being searched.
VB Code can select the table to be searched by changing the RecordSource property of the data control to a valid SQL string or table name. If this property is changed at runtime, all bookmarks in the DataFind control should be set to empty strings. The code should also deal with any errors caused by invalid table names etc., which may cause a run time error.
The above parameters can contain control names in order to substitute previously obtained values. The record found by the DataFind control can be used by DataGet controls to access particular fields in the record.
Example:

```
Dim newbookmark as string
DataControl.Recordset.FindNext mycrit
DataFind1.Bookmark(channel) = DataControl.Recordset.Bookmark
```

Result

Applies to DataGet dialog, IniSw , DataNew

This Result property is a string containing the value of the field found by the last search. It can be read and set by VB code in the Enter or Error event before branching takes place.

Example 1
To access the result:

```
mystring = DataGet1.Value(channel)
```

Example 2
To override the result:

```
DataGet1.Value(channel) = this is the result I really wanted
```

DataSource

(data control name)
Applies to DataFind dialog, Record , DataNew

This field contains the name of the data control that defines the data to be acted upon. In PlayMsgs and Record controls, this is the name of the data control that defines the database containing the message tables. using the DataBaseName and RecordSource properties in the data control. The data control used by a DataFind control can be shared with other DataFind and DataNew controls, but not with Record or PlayMsgs controls.
Likewise Record and PlayMsgs controls can share data controls with each other, but not with DataFind and DataNew controls.
This property should not be changed while the voice system is running.

RecordCount

Array Integer(1, MAXCHANNELS)

Applies to DataFind dialog

This field contains the number of fields found in the database that match the given specifications. The property is set to 0 when a call enters the control via the main input, and increments by 1 every time a record is found. If the Calculate Record Count on entry box is checked in the setup box, the record count is calculated as soon as a control enters via the main input. Counting fields can be time consuming in a large database, and should be avoided if possible.

Example 1: setting the RecordCount

If using your own search algorithms, you can set the RecordCount yourself:

```
DataFind1.RecordCount(channel) = DataControl.RecordSet.RecordCount
```

Example 2- reading the RecordCount

You may want to execute some code based on the value of RecordCount:

```
if (DataFind1.RecordCount(channel) = 0 then
        User.GotoNode(channel) = 0;
Endif
```

DataFind dialog

See Also DataFind dialog

Field Specification

Search in database...
This is a list of the available DataFind controls on this form. The DataFind control you choose together with the VB data control it links to should be set up before filling in this dialog.

for ...

All records
If this button is checked, DataFind will select the next record on each entry. The data matching fields will not be visible when this option is checked.

Records containing
If this button is checked, DataFind will search the database for the next record that matches the specifications in the *Records containing* text field and Match type specifications.
The text you enter may contain control names.

Find Control

This command creates or shows the Control List window. You can choose a control name from this window and copy it into the clipboard with the Copy button. The control name can then be pasted into any text field as a parameter using Shift+Insert. Only controls on loaded forms that have default properties are listed in the window. If you have a large number of controls, you can quickly find the control you are looking for by selecting the form name first to show only the controls on that form.

in field...
If you have set the database field to a VB data control which has a valid DatabaseName and RecordSource, this field will show a list of fields in the selected database. Select the field that you wish to search.

Match fields

Exactly
This search type will do a normal comparison, character by character, between the database contents and the data in the *For...* field.

Use digit translation
This search type will translate the text in the database field to the telephone keypad equivalent before performing the comparison. This is useful for directory lookup, etc., where the caller is limited to entering telephone keys in order to enter an alphanumeric name.

Match on

The DataFind control can do two kinds of matching:

First characters
This search algorithm will match the *Records containing* text field against the database contents. It will stop matching and report a successfule match if it reaches the end of the *Records containing* text field without a mismatch. See Data search specification.

All characters
This will match the entire *Records containing* text field field against the database contents. In this case the database field and the *For...* field must be exactly the same.

Calculate Record Count on entry

This field can be checked if you wish to have the DataFind control count the number of records as soon as a call arrives in the control. This may be useful if you wish to announce the number of items in a database before allowing the caller to choose one.

Use global bookmarks

Check this button if you want all channels to use the same record pointer. Normally, each channel will maintain its own position in the database. For instance the caller on channel 1 may be on record 3, and when re-entering the DataFind via the Next input, will search again from record 3. At the same time, the caller on line 4 might be on record 1232, and will restart his search from that record. If you check Use global bookmarks, there is only one database position pointer. This is useful in outcalling applications, where a number of channels are dialing a list of numbers, and you want each number dialed once regardless of which channel does the dialing.

Properties

- GotoControl
- GotoNode
- Nodename
- Result

Events

- Condition
- Enter
- Exit
- VoiceError

DataGet control

 Properties Events

DataGet dialog
Data Condition dialog

Default Property: Result

Description:

When a call enters the DataGet control, it reads a field from the database record found by a previous DataFind control and stores the result in the property 'Result'. The call is then passed to the control connected to one of the output nodes, based on a comparison operation performed on the field value obtained.
The field name and comparison conditions may contain calculated values from other controls. The Result property can be used by other controls, and VB code.

Pattern Matching

The DataGet control can contain up to 8 different patterns against which it can test the data found. Pattern matching can be performed exactly, or using a digit to alphabetic character translation. Translation is useful when matching collected digits entered by the letters on the telephone keypad against names. In addition, the DataGet control can perform complete matching, or it can use a find-first method which will match any field that begins with the characters given. For instance, a condition "JO" will match fields containing both "JOHN" and "JONAS" in the database.
If the record is found, but there is no match between the data field and any of the conditions, the call is transferred to the "No Match" node. If a match is found between the value obtained and one of the result fields, the call is routed to the appropriate node.
You can also bypass or add to the DataGet condition matching by checking the Let VB decide check box in the decision dialog. This will generate a VB event for that condition to allow your own code to check the match.

Using with DataFind

The DataGet must be connected to a DataFind control either directly or via one or more DataGet, IniSw or Count controls.

Result

Applies to DataGet , IniSw

Array String(1, MAXCHANNELS)

This Result property is a string containing the value of the field found by the last data access or ini file read operation. It can be read and set by VB code in the Enter or Error event before branching takes place.

Example 1

To access the result:

mystring = DataGet1.Value(channel)

Example 2

To override the result:

DataGet1.Value(channel) = this is the result I really wanted

NodeName

Array String(0, Number of decisions-1)

This property contains an array of strings containing a name for each node. This property is read-only.

Condition Event

Applies to DataGet , GetDigits

This event is generated when you specify Let VB code decide in the condition dialog. Digit collection will terminate when the maximum digits or a a terminating digit have been received, or digit collection has timed out. The condition event is defined as
Sub xx_Condition (channel as Integer, decided as Integer, node as Integer);

If your code determines that there is a match for this output node (as specified by the node parameter), it should set the decided parameter to TRUE and exit. Otherwise pattern matching will continue on the next node. If your code changes the node parameter in addition to setting *decided* TRUE, the call will be transferred to the node number you set. Nodes are counted from 0, starting from the top node.

DataGet control:
You can access the node name and the result using the Nodename and Result properties.

GetDigits control:
You can access the node name and the digits using the Nodename and Digits properties.

Data Condition dialog
Digit Condition dialog

DataGet dialog

See Also DataGet

Get data from ...
This is a list of the available fields in the database that the DataGet is linked to.

in (control name)
This label shows the name of the DataFind control that this control is connected to. This DataFind control in turn links to a VB data control that specifies the database file to access.

Compare data using...

Specify here the kind of comparison that you want to perform between the database contents and the condition fields.

Exact Match
This search type will do a normal comparison, character by character, between the database contents and the data in each condition field.

Alpha digits
This search type will translate the text in the database field to the telephone keypad equivalent before performing the comparison.

Match On

The DataGet control can do two kinds of matching:

All characters
This will match the entire database field against the condition field. The database field and the condition field must be exactly the same.

First
This search algorithm will stop matching as soon as it reaches the end of the characters in the condition field.

Test Conditions

Edit command
Use the Edit command, or double click on an entry to show the Conditions dialog.
Delete command
Deletes an entry
New command
Use this command to create a new condition.
See Data Condition dialog

Data Condition dialog

See also DataGet

Let VB code decide..

Check this box if the pattern matching capablility in DataGet does not meet your requirements. For every node that has this button checked, the DataGet control will generate a Condition event in the control, allowing your code to decide if the digits meet your criteria for acceptance. See Condition event above.

Database Field Value

Sets the value required in the database field to make the call exit vai this node.

Condition Name

The name of the condition that is drawn in the node which owns this condition.

Find Control

This command creates or shows the Control List window. You can choose a control name from this window and copy it into the clipboard with the Copy button. The control name can then be pasted into any text field as a parameter using Shift+Insert. Only controls on loaded forms that have default properties are listed in the window. If you have a large number of controls, you can quickly find the control you are looking for by selecting the form name first to show only the controls on that form.

DataNew Properties

- DataSource
- GotoControl
- GotoNode

DataNew control

Properties Events

Default Property: None

DataNew dialog

Description

Adds a new, blank record to a database. The database is specified by the DataSource property, which can be changed in the setup dialog. Use a DataChange control connected to the DataNew control to set the fields in the new record. The data control used by a DataNew control can be shared with other DataNew and DataFind controls, but not with Record or PlayMsgs controls.

IMPORTANT NOTE:
There is a bug in Visual Basic 3.0 which prevents the addition of a new record to an empty table. To circumvent this, ensure that tables are created with a dummy record before using them in Visual Basic.

DataNew dialog

See also DataNew
The DataNew dialog has only one field, Database Specification which selects which VB data control to update with a new record. When a call enters this control, a new blank record will be added to the database.

Properties

- GotoControl
- GotoNode
- StopDelay

Events

- DelayStarted
- Disconnect
- Enter
- Exit
- PhraseError
- PlayRequest
- VoiceError

Delay control

 Properties Events

Default Property: None

Delay dialog

Description:

Implements a wait period. Delay controls can be used if your VB code has to perform a lengthy activity, and you want the caller to wait until it is complete, for instance to wait for a busy line to be free before re-attempting a transfer. The Delay control can operate in three modes:
Simulated Hold: VBVoice plays hold music to the caller during the wait period.
Hold: VBVoice puts the caller on hold using the PBX hold feature, as defined by the PUTONHOLD and RECONNECTFROMHOLD dialing strings, and hangs up the phone while waiting.
NOTE: This mode should only be used if there is no possibility of another incoming call on this line.
Silent: VBVoice remains silent until the delay is complete.
When playing hold music, VBVoice continually plays a short file containing music or a short announcement. This can be replaced with your own phrase if required. The delay control checks the delay time after each completed play operation, so the voice file to be played should not be very long.

StopDelay

Applies to Delay

Array Integer(1, MAXCHANNELS)

Set the StopDelay property to zero at runtime during a wait period will allow the call to continue.

Example

DataGet1.StopDelay(channel) = 0

DelayStarted

Applies to Delay

This event occurs after the caller has been put on hold. Use this event to initiate any lengthy activity required, rather than the Enter event. Yielding is still required to allow other channels to run. (See Properties and Events). If you start a lengthy activity in the Enter event, the caller will not be put on hold , since the Enter event will not return to VBVoice for that channel, even if your code yields to Windows.

Delay dialog

See also Delay

Put on hold during wait

Music during wait

Silence during wait
These three buttons select the action to be taken during the wait period.

Delay time
Sets the period of the delay, in seconds.

Music file
Specifies a file to be played during the delay, when Music during wait is selected

Properties

- CPLFile
- GotoControl
- GotoNode
- NumToDial

Events

- Disconnect
- Enter
- Exit
- PhraseError
- PlayRequest
- VoiceError

Dial control

 Properties Events

Default Property: None

Dial dialog

Description:

The Dial control can be used to go offhook and start a call, perform a transfer to another extension, or to dial some digits during a call. If the number provided number is prefixed with the letter 'S', call supervision is enabled. The number can contain control names, which will be replaced by the value of the indicated control before dialing.
Transfer
If Use Transfer is set, the Start Transfer digits will be dialed before the supplied string. The Dial control will then wait for dialtone before dialing the digit string specified in the dialog. If no dialtone is found, the control will dial the Reconnect from busy... digits and exit via the NoDialtone output.
Otherwise the call will dial the digits and proceed as follows:

Supervised transfer
If supervision is enabled, the control waits until a time-out or a known telephone condition is detected on the line, such as answer, busy or fast busy.

* Answer: The call proceeds while connected to the new transferee, and the original caller is kept on hold. The original can be reconnected using the Complete transfer after answer string in a subsequent Dial control.
* Busy, No Answer, Fast Busy: The original caller is reconnected using the Reconnect from hold... string, and the call continues with the original caller, in the control connected to the output node as defined by the call progress type.

Unsupervised transfer
If supervision is not enabled, a blind transfer will be performed by dialing the transfer digits and the number. After dialing, the call will be terminated.

No Transfer supervised dial
The line is taken off-hook, if not already off hook and the control waits for dialtone before dialing the digits. Once the digits are dialed, the control waits for a recognized call progress condition (answer, busy etc.) and exits from an output node depending on the call progress result. This mode can be used to start a call.
If the control is being used to initiate a call, note that it is possible for a call to arrive just as the call control is going off-hook. In this case, the call will continue via the NoDialTone output.

Unsupervised (blind) dial
This mode can be used to dial some digits during a call, for instance to send some digits to another voice response system.

The digits are dialed and the call is transferred to the control connected to the Dialled output.

NumToDial

Applies to Dial

Array String(1, MAXCHANNELS)

The dialed number is recalculated for each call if it contains control names. The NumToDial property makes the calcualted number available to VB code in the Enter event, and code can also set a new number. This does not change the number for subsequent calls..

Example 1

To access the number:

```
Dim actualnumber as String
actualnumber = Dial1.NumToDial(channel)
```

Example 2

To override the predefined number with your own:

```
Dial1.NumToDial(channel) = 613-839 0033        spaces, hyphens and brackets are allowed
                                                max string length is 64 characters
```

CPLFile

Applies to Dial

Array String(1, MAXCHANNELS)

The CPLFile property defines the set of parameters to be used when determining call progress conditions - busy, ringback, no answer etc. You may need to specify a different file for in-house PBX calls and for outside lines, due to differences between the ringing and busy tones supplied by the PBX, and those received from the public network. Normally this can be done using a database entry to specify the file, however it can also be set by code in the Enter event. When the CPLFile property is changed at runtime, the change only applies to the call that caused the event.

Look in the Getting Started booklet for your voice card for how to create CPL files and to configure call progress analysis for your card. The CustomCPL check box must be set in order for this property to take effect. The structure of the CPL file is described in the .BAS definitions file for your voice card. The CPL file can also be created using the MAKECPL utility. Some voice cards have their own utilities for defining call progess analysis.

Dial dialog

See also Dial

Number to call

This field is used to enter the number that this control should dial. The field can contain control names, for instance it can refer to a DataGet control which will cycle through a list of numbers to dial. If the number begins with S, or a control name is used, the Call Progress Analysis field is made visible. A runtime error will occur if a number containing invalid characters is detected.

Use Transfer feature
Check this button if you want VBVoice to transfer the current call to the number entered. See *Transfer* above.

Call Progress Analysis

Use Custom CPL file / Filename
Some voice cards cannot handle both internal PBX calls and outbound (public telephone system) call progress at the same time. Use this field to select whether to use the default CPL file (VBVOICE.CPL), or a different one. In this way you can select which set of call progress parameters to use for each call. Creation of CPL files is specific to the voice card you are using. Refer to the Getting Started booklet for your voice card for details on how to create CPL files.

Find Control

This command creates or shows the Control List window. You can choose a control name from this window and copy it into the clipboard with the Copy button. The control name can then be pasted into any text field as a parameter using Shift+Insert. Only controls on loaded forms that have default properties are listed in the window. If you have a large number of controls, you can quickly find the control you are looking for by selecting the form name first to show only the controls on that form.

Properties

- Digits
- GotoControl
- GotoNode
- MaxSil
- MaxKeys
- Nodename
- TermDtmf

Events

- Condition
- Disconnect
- Enter
- Exit
- PhraseError
- PlayRequest
- VoiceError

GetDigits control

 Properties Events

Default Property: Digits

GetDigits dialog
Digit Condition dialog

Description:

The GetDigits control plays a greeting and waits for digits from the caller. Pattern matching is performed on the collected digits using up to 14 exit conditions. GetDigits provides built-in invalid digit and time-out handling. The GetDigits control can also be used just to play a greeting and ignore all digits.

Digit Pattern Matching

The termination conditions can be as simple as a single digit per exit branch (the default configuration) or complex sequences consisting of the following characters:

$	any char or series of characters. Use a terminating digit, maximum chars or silence to terminate input.
n	any single numeric char 0 - 9
- (hyphen)	specifies a numeric range - i.e. 222-333.
?	specifies any one char
0-9 *#	each character specifies an actual digit to be received.
VBEvents	In addition, you can specify that a VB event handler should decide if the digit pattern is correct. See Condition Event. Used in the Voicemail example design to check the callers new password before setting it in the database.
Control Names	Conditions can also contain control names. This is sometimes useful when comparing digits recieved against database values. This method is used in the VoiceMail example to check the subscribers password.
Wild Chars	If using a wild character as a termination in conjunction with other termination conditions, put the wild character in the last digitmask in the list. The digitmask list is processed from top to bottom after the digits are collected, so the wild character may hide other valid digitmasks that should take priority.

Play Termination

Normally GetDigits will allow the experienced caller who knows the system to bypass the greeting by pressing one or more digits. However you may have important information that you want the caller to hear. You can do this by clearing the 'Allow play termination on digit' checkbox in the setup dialog. If this flag is cleared, the greeting will always play to completion and any collected digits will be cleared before starting to wait for digits.

Wait for Digits Termination Conditions

The GetDigits control automatically calculates the required termination conditions for collecting digits. It can select termination based on maximum number of digits, termination characters, or silence. A warning will occur on startup if no termination except silence is possible. Silence, a terminating digit or a maximum digit count must be used to terminate digit collection, if:

- any condition contains the wild character $
- any condition contains a Control name
- a VB Condition is used

In these cases VBVoice has no other way of detecting whether the caller has entered all the digits.
If silence is the only possible termination, the response time for the caller will be longer, and type-ahead will not be possible. The system has to wait for the silence time-out period before determining that the

caller has stopped entering digits,.
You can avoid using Silence with the Max Digits or Always Terminate On digit setting. In most systems, # can be used as a termination digit to avoid using time-outs. If a termination digit is set using this field, the digit will be stripped from the received digits before continuing, i.e. the Digits property will not contain the terminating digit even if received. If a terminating key has been set, and it is required to set a valid condition where only the termination digit is entered with no other digits, use a blank digit mask in the condition field.

Type-ahead and silence timeout terminations

Note that if a GetDigits control terminates with a silence timeout, all digits collected will be consumed by that control. Even if the GetDigits matches on a field with 2 digits, and 3 digits have been entered, the extra digit will not be available to subsequent controls for type-ahead operations.

Invalid digits, No digits handling

If an invalid digit sequence is entered by the caller, i.e. a sequence that does not match any of the digit masks, or if no digits are entered, the control allows the user to try again up to a preset number of attempts. A special prompt can be set for either or both of these events. If the error count exceeds the number of retries set when invalid digits or a no digits event occurs, the GetDigits control will perform error handling as described below.
If the error count has not been exceeded, and an invalid digit sequence has been received, VBVoice will play the invalid digits greeting, if set, followed by the entry greeting, and increments the error count. The default Invalid Digits Greeting is that was not a valid entry. If no digits have been entered, VBVoice will play the No Digits greeting, followed by the initial greeting, and increments the error count. There is no default No Digits Timeout greeting.

Err input

The GetDigits control can also be entered from a second input "Err". This input causes the control to act as if invalid digits had been received: it increments the error count, and if the maximum retry count is exceeded, the call is transferred to the 'Invalid' exit node. This input can be used when digits have been collected and are then checked against a database. If the digits are incorrect, they can be treated the same as digits checked by the control itself.

MaxSil

Applies to GetDigits

This is the maximum duration of silence (in seconds) while waiting for digits before issuing a time-out. This property is set by the control to the value of *Maximum Silence* in the setup dialog. It can be changed in the Enter event to another value if required. The new value will only affect the current call.

Example
To override the maximum silence setting:

```
If CallerName = Fred Then
        GetDigits1.MaxSil(channel) = 5  stop after 5 seconds silence
                                        Fred is a slow talker
Endif
```

MaxKeys

Applies to GetDigits

The maximum number of keys to accept. This value is set by the control to the default Max Keys set in the setup dialog, or a value based on the digit masks entered. It can be changed in the Enter event to another value if required. The new value only affects the current call.

TermDtmf

Applies to GetDigits

This is a bit-mapped array indicating the keys that will terminate the digit collection. This property is calculated at startup based on the digit conditions in the control. It can be changed in the Enter event to another value if required. The new value only affects the current call. Only one digit in the mask is allowed with Pika cards. Use the constants MASK_0, MASK_1 etc. defined in VBVOICE.BAS

Example
To override the TermDtmf property:
GetDigits1.TermDtmf(channel) = MASK_0 OR MASK_S terminate on 0 and * only

Digits

Applies to GetDigits

This property contains the digits collected by the control. This is the default property.
It is sometimes useful to override the result of the GetDigits control: if you have two paths that both collect the same number - i.e you may be asking for a mailbox number, and provide a method of direct input, or a method using database lookup. Subsequent code must reference one control only for the result, so you can use the exit event in say the database loopup path to set the Digits property for the direct path as if the caller had entered the digits directly via than by database lookup. This method is used in the VMDEMO program, in the DIRECTRY form.

Example 1
To access the result:
mystring = GetDigits1.Digits(channel)

Example 2
To override the result:
GetDigits1.Digits(channel) = this is the result I really wanted

GetDigits dialog

See also GetDigits

Termination conditions

Max digits
This field sets the maximum number of digits that can be received before GetDigits terminates digit collection.

Maximum silence
This field specifies the number of seconds that GetDigits will wait for a digit. If a digit is not received in the time, digit collection will be terminated. If this field is set to 0, GetDigits will not wait for a digit after the Greeting has been played, but will proceed immediately with digit processing.

Number of retries on error
This field specifies the number of invalid digits, and silence time-outs that can occur before GetDigits passes the call to the Silence or Invalid nodes, or invokes the global error handler.

Retry on silence
If this box is checked (the default), both silence timeout and invalid digit events use the retry count. If this box is unchecked, then a silence timeout will cause the call to exit via the silence output immediately, regardless of the number of retries set. Invalid digits increment the retry count as usual. This is useful in initial menus where you want to give the caller several attempts to enter the correct digit, but want to allow callers without touch-tone phones to reach the operator.

Always terminate on
This field specifies a digit that can be used to terminate digit collection. This often speed up menu selection by allowing the caller to enter say the # key to end a sequence of digits, rather than having to wait for a silence time-out.

Routing

Edit command
Use the Edit command, or double click on an entry to change an exit condition using the Edit Conditions dialog (see Condition dialog).

Delete command
Deletes an exit condition.

New command
Use this command to create a new exit condition.
Digit Condition dialog

Clear digits on entry

Set this button if you want to clear all previously collected digits from the VBVoice digit buffer. Normally you will want to allow experienced callers to save time by typing ahead since they know which order the prompts are played, and what digits are required.

Allow play termination on digit

If you have a message to play the caller, which you really want them to hear, check this button. See *Play Termination* above.

Error Handling

Invalid digit prompt

No digits prompt
These two buttons allow you to create a custom greeting to be played after the caller enters invalid digits,

or no digits at all. After the error greeting is played, the initial greeting is played again, and GetDigits starts collecting digits again.

Digit Condition dialog

See also GetDigits

Let VB code decide..

Check this box if the pattern matching capablility in GetDigits does not meet your requirements. For every node that has this button checked, the GetDigits control will generate a Condition event in the control, allowing your code to decide if the digits meet your criteria for acceptance. See Condition event above.

Digit Mask

Sets the acceptable digits to allow the call to exit this node. See Digit Matching.

Condition Name

The name of the condition that is drawn in the node which owns this condition.

Find Control

This command creates or shows the Control List window. You can choose a control name from this window and copy it into the clipboard with the Copy button. The control name can then be pasted into any text field as a parameter using Shift+Insert. Only controls on loaded forms that have default properties are listed in the window. If you have a large number of controls, you can quickly find the control you are looking for by selecting the form name first to show only the controls on that form.

Properties

- GotoControl
- GotoNode

Events

- Disconnect
- Enter
- Exit
- PhraseError
- PlayRequest
- VoiceError

Greeting control

 Properties Events

Default Property: None

Description:

The Greeting control plays a greeting and exits via the Done output.

Play Termination

Normally Greeting will allow the experienced caller who knows the system to bypass the greeting by pressing one or more digits. However you may have important information that you want the caller to hear. You can do this by clearing the 'Allow play termination on digit' checkbox in the setup dialog. If this flag is cleared, the greeting will always play to completion and any collected digits will be cleared after the play is completed.

Play Option Digits

These digits allow the caller to control the playback of a greeting by jumping ahead or moving back to replay a section. The greeting must consist of a single phrase, either a VAP phrase, VOX file, or a System phrase of type VAP Phrase by Index, VAP Phrase by Name or File Spec. The default settings for the times may be changed in VBVOICE.INI, section [Voicecard], items SkipFBytes and SkipBBytes.

Interaction with Play Termination on Digit option

Note these digits will work regardless of Play Termination on Digit. If play termination on digit is set, and an option digit is received, the option is performed. If another digit is received, the play terminates. If play termination is not set, and a digit other than an option digit is received, it is ignored.

Fast Forward

This digit will skip the playback forward when received The default time is 1 second.

Rewind

This digit will rewind the playback backwards when received. The default time is 2 seconds.

Pause

This digit will terminate the playback. The playback will restart at the same position when another digit is received, or after 20 seconds.

InCon

 No Properties No Events

Description:

Used for connection from an output on one control to the input of another on a different form. Other controls cannot be connected across forms.

Properties

- GotoControl
- GotoNode
- Result

Events

- Enter
- Exit
- VoiceError

IniSw

Properties Events
Default Property: Result

IniSw dialog

Description:

Searches for a entry in a Windows initialization (.INI) file, and routes the call depending on the result found. You can set the filename, section name and field name of the entry you wish to test. The result property is set from the data found. The filename, section name, entry name and tested conditions can all contain control names. Pattern matching will be performed on the collected digits using up to 8 exit conditions in the same way as in the DataGet control, using find first or exact matching, and optional numeric to alphabetic translation.

Call Routing

If the ini file entry is not found, the call is routed via the "Not Found" node.
If the entry is found, but there are no conditions set, the call is transferred to the "Found" node. If conditions are set, but there is no match between the data field and any of the conditions, the call is transferred to the "No Match" node. If a match is found between the value obtained and one of the conditions, the call is routed to the appropriate node.
Otherwise the call is routed to the "Default" node.

IniSw dialog

See also IniSw

Branch on

Field

Section

File
These three fields specify the Windows .INI file, the section within the file (this is the field contained within the [..]) and the field name. The INI file must be in the Windows directory or have a full pathname.

Test Conditions

Edit command
Use the Edit command, or double click on an entry to change an exit condition using the Edit Conditions dialog (see page ...).

Delete command
Deletes a exit condition.

New command
Use this command to create a new exit condition.

Find Control

This command creates or shows the Control List window. You can choose a control name from this window and copy it into the clipboard with the Copy button. The control name can then be pasted into any text field as a parameter using Shift+Insert. Only controls on loaded forms that have default properties are listed in the window. If you have a large number of controls, you can quickly find the control you are looking for by selecting the form name first to show only the controls on that form.

Events

- Enter
- PhraseError
- PlayRequest
- VoiceError

OnHook control

 No Properties Events

Description:

Plays a greeting, and hangs up the phone. Control of the line is returned to the Phone control that owns this channel, where a Disconnect event will occur. This event will allow termination code to run for this call.

OutCon control

 N**o Properties** **No Events**

Default Property: None

Description:

Used for connection from an output on one control to the input of another on a different form. Other controls cannot be connected across forms.

Properties

- CallerId
- Channel
- MaxTime
- Mode
- RingsToAnswer

Events

- Disconnect
- Exit
- Ring
- VoiceError

Phone control

Properties Events
Default Property: CallerId (with CallerID option)

Phone dialog

Description:

This control owns a telephone line, and can receive calls and initiate calls on this. The channel number can be set in the dialog at design time, or the channel can be assigned at run time via the channel property.

Operating modes

Operating Modes

This control can operate in 3 modes:

Wait for ring
In this mode, the control waits for the number of rings set by RingsToAnswer to be received. It then takes the line off-hook and passes the call to the next control. A Ring event is generated in the control before the line goes offhook to allow your code to perform call initialization and/or to override normal behavior. If the optional CallerId module is purchased, the caller id is available from the caller_id property. If RingsToAnswer is set to zero (blank) the control will not detect ringing.

Start call after delay
In this mode the call is started as soon as the system starts, or when the previous call on the channel owned by this control is terminated. A delay is used after the termination of the previous call to ensure that the line is detected as on-hook by the telephone system before starting the next call. If ringing is detected during the delay time, the Phone control will wait 20 seconds for the line to clear before starting the next call. The next voice control in the chain will normally be a Dial control to start an outgoing call.

Idle
The control does not respond to Ring events or perform any action until the mode is changed.

Channel Property

Applies to Phone

This property is normally set at design time, but can be set at run time. If the channel property is set to 0 (or left blank) at design time, no channel is allocated. If the channel property is set to 0 at runtime, VBVoice will attempt to find a free line. If successful, the channel property will then contain a valid line number. Lines are numbered from 1 up to the maximum lines available.
If unsuccessful, an error will be generated.

Example 1
To access the channel
myinteger = Phone1.Channel

Example 2
To set the channel:
Phone1.Channel = 5

RingsToAnswer Property

Applies to Phone

Defines the number of rings required before answering a call. May be set to 0 (blank) to inhibit call answering.

Example
To set the RingsToAnswer:
Phone1.RingsToAnswer = 5

CallerId Property

Applies to Phone

String

Only available with caller id option. Provides the calling number as received by the CallerId hardware. If Caller Id is not configured, the property is not available. The property can also be set by VB code.

Example
To access the caller id:
mystring = Phone.CallerId

Mode Property

Applies to Phone

Integer

Can be set to one of the five constants, corresponding to the three operating modes and also 2 actions. The operating mode constants can be ORed together.

If mode is set to STARTCALLNOW from *idle* or *wait for ring*, a call will be initiated and the mode will revert to *idle* at the end of the call.

GLOBAL CONST IDLE = 0
GLOBAL CONST STARTCALLNOW = 1
GLOBAL CONST CALLAFTERDELAY =2
GLOBAL CONST WAITFORRING = 4
GLOBAL CONST STOPCALLNOW = 8

Example
Phone1.Mode = STARTCALLNOW

When the mode is set to STARTCALLNOW, a call is started in the same way as CallAfterDelay (desribed above), except the call is started immediately. StartCallNow does not take the phone offhook - use a Dial control to take the line off-hook and dial a number. If the mode to set to STARTCALLNOW while a call is active on the line, a runtime error will be generated.
If the mode is set to CALLAFTERDELAY, a call will be started after the delay period from when the line becomes idle, if not already idle.

Example

Phone1.Mode = STARTCALLNOW

When the mode is set to STOPCALLNOW, the control behaves in the same way as described in the MaxTime property. The mode property can be set to STOPCALLNOW at any time during a call.

Disconnect Event

Applies to All Controls

When a call that was initiated by this control is terminated, for whatever reason, a Disconnect event occurs in this control. The Disconnect event occurs just before the system hangs up the phone. This allows call termination code to run.

```
Sub Phone1_Disconnect (Channel As Integer, Reason As Integer)
```

The reason for disconnect can be one of the following constants:

```
Global Const CONTROLHANGUP = 0  the onhook control hung up the line
Global Const SYSERRORHANGUP = 1         a system error occurred
Global Const CALLERHANGUP = 2           the caller hung up: a loop current drop or other
disconnect indication was received
Global Const INVALIDHANGUP = 3     invalid digit or silence timeout handler was invoked
```

Ring Event

Applies to Phone

The Ring event occurs when the number of rings specifed by RingsToAnswer have been received, but before the Phone control takes the line offhook. A Ring event can occur regardless of which mode the control is in, but if RingsToAnswer is set to 0, no ring events will occur. If *NoOffhook* is set to TRUE, the phone will not be taken offhook. The default is FALSE - the line is taken off hook.

Sub ControlName_Ring(Channel as Integer, NoOffhook as Integer)

Phone dialog

See also Phone

VoiceLine
Specifies the voice card channel to be allocated in this control. This field can be left blank if you want to allocate a line at run-time.

Channel Mode

Allocate Line
Not used

Answer Call
The Phone control will wait for the number of incoming rings specified in Rings before Answer, then go offhook and transfer the call to the control connected to the Ring node.

Start call immediately
The Phone control will start a call and transfer it to the control connected to the StartCall node.

Line Type

Loopstart
This is the normal telephone line supplied to single line telephones.

T1 Winkstart
Use this line type when using the Rhetorex MDTI T1 interface card.

Loopstart DID
Use this line type to have VBVoice automatically collect incoming digits from DID lines when using an external DID trunk connection device such as those from Exacom and VoicePRO Systems. The digits are available in the CallerID property.

Rings before answer
Specifies the number of rings to be received before the Phone control will answer the call.

Delay
When in Start call immediately mode, specifies the number of seconds to wait from the end of one call to starting the next. Normally 4 seconds should be used to ensure that dialtone is received.

MaxTime
When set to a non-zero value, interrupts a call at the time specified.

Properties

- GotoControl
- GotoNode

Events

- Disconnect
- Enter
- Exit
- MsgUpdate
- PhraseError
- PlayRequest
- VoiceError

PlayMsgs control

Properties Events
Default Property: None

PlayMsgs dialog

Description:

A high level control that plays messages from a database containing message tables. It is the main engine required to implement a voice mail system using VBVoice.

Operation

Two types of messages are supported, New and Old. New messages are those that have not yet been heard, or have been saved as new. New messages are converted to Old once they have been heard. When a call enters a PlayMsgs control, it first plays the number of each type of message, and prompts the caller to select which type of messages (New or Old) to hear, if there is a choice. It then plays the messages in sequence. After each message a configurable options greeting is played, allowing the caller to save, delete, forward or skip the message, or exit. After both groups of messages have been played, the control starts again at the beginning. If there are no messages left, the call exits the control. Once a new message has started playing, it is changed to an old message unless the caller selects save-as-new, delete, skips the message or hangs up before the end.
(Message forwarding not implemented yet).

Message Database

The PlayMsgs control uses text-based files to store the message database. Each mailbox uses 2 files, one for current messages (new and old), and another which lists messages to be deleted. The mailbox filename is BOX<mailbox number>.MSG

Database File format

Each current message file contains 2 headings: [New] and [Old]. Under each heading are the filenames of the messages, and an optional source mailbox number separated by a tab character. When files are heard by the mailbox owner, they are moved from the new list to the old list. If they are deleted by the mailbox owner, the filename is removed from the current message file and added to the deleted file list The mailbox filename is BOX<mailbox number>.DLT.
The message filename is calculated when the call enters the PlayMsgs control and is normally selected by prefix letters concatenated with the mailbox number. Example: BOX%MailBox%.MSG where the tables are named BOX100 to BOX 999, and a GetDigits control called MailBox has collected 3 numeric digits as the mailbox number. The filename is the path of the message file, relative to the main VBVoice directory. It can be in a subdirectory if required.

Table Name Checking

The mailbox number should be validated before the PlayMsgs control tries to access the message file. Validation can be achieved using a separate mailbox database of valid mailboxes, which can be used to check the mailbox number before entering the PlayMsgs control. This table can also contain the password and other mailbox parameters. This is the method used by the sample voice mail form in the VOICMAIL directory. See the voicemail example code and description (Users Guide, page **Error! Bookmark not defined.**) for more details of how to use the PlayMsgs control.

Option Digit Handling

Delete

Marks the current message as deleted. The file is not erased but the filename is moved from the current messages file to the deleted messages file. An off-line utility is responsible for deleting files. This allows for the possibility of undeleting deleted messages.

Save as New

Marks the current message as a new, unheard message, by moving the filename into the [New] list.

Skip

Skips to the next message. The status of the current message is not changed.

Info
Plays the time and date of the message.

Forward
Forwards the message to another box.

Exit
Exits the PlayMsgs control.

Review
Replays the current message from the beginning.

Next
Moves to the next message, marks the current message as old.

Invalid digit, No digit handling

If an invalid option digit is received, VBVoice will increment the error count, and play the error prompt and the options greeting again. If silence is detected - i.e. the caller did not not press any digits, VBVoice will increment the error count, and play the no digits prompt followed by the options greeting. When the error count exceeds the maximum error count, the call is transferred to the control connected to the NoMsg node, or if this node is not connected, VBVoice continues with default error handling. The error count is reset when the caller presses a valid key.

Prompts used:

There are three greetings used in this control. The VAP file PLAYMSGS.VAP contain some special phrases used by these greetings. The phrases are indexed by position, rather than name, so the order of the phrases should not be changed.

Options greeting:
This will normally say "Press 1 to save your message, 2 to save your message as new, 3 to delete your message, 4 to hear the next message, or 8 to exit". This greeting should be changed to match the digit options you have set.

Extra Prompt after error.
This prompt will inform the caller that they have pressed an incorrect digit. The options greeting will then be played again, until the error count is exceeded. The default greeting is that was not a valid entry.

Prompt after no digit.
This prompt is played if no digit is pressed during or after the options greeting. The options greeting will then be played again, until the error count is exceeded. There is no default Silence Timeout greeting.

PlayMsgs dialog

See also PlayMsgs

Play Option Digits

Delete

Save as New

Skip

Info

Forward

Exit

Review

Next
These selection boxes allow you to specify which options should be available to the caller at the end of playing each message. To disable the option, set to digit to none. The Options prompt should be set to match the options specified. See *Option Digit Handling.*

Options Prompt
Use this button to create the greeting that tells the caller what message options are available after playing each message.

Error Handling

Invalid digit prompt

No digits prompt
These two buttons allow you to create a custom greeting to be played after the caller enters invalid digits, or does not enter a digit. After the greeting is played, the initial greeting is played again, and GetDigits starts collecting digits again

Message Files

Mailbox Number
This field specifies which mailbox to use. Normally this field refers to a previous GetDigits control which has received the mailbox from the caller. This number is used to find the message files.

Find Control

This command created or shows the Control List window. You can choose a control name from this window and copy it into the clipboard with the Copy button. The control name can then be pasted into the File path field as a parameter using Shift+Insert. Only controls on loaded forms that have default properties are listed in the window. If you have a large number of controls, you can quickly find the control you are looking for by selecting the form name first to show only the controls on that form

OptionDigit

This property can be used after a call has exited the Record control, to access the last option digit entered by the caller.

Invalid

This event occurs when the caller enters a digit which is not a valid option digit. VB code can choose to transfer the call to another control based on the digit selected (using the GotoNode, Gotocontrol properties) or allow the Record control to treat it as an invalid digit in the normal way. The digit is passed to the procedure allowing your code to decide what to do.

Sub Record_Invalid (Channel As Integer, Digit As Integer)

Properties

DataSource

Filename

GotoControl

GotoNode

MaxSil

MaxTime

OptionDigit

Events

- Disconnect
- Enter
- Exit
- Invalid
- PhraseError
- PlayRequest
- VoiceError

Record control

 Properties Events
Default Property: None
Record dialog

Description:

Records a file into the filename specified, and can update a database table, or a message file in the format required by the PlayMsgs control. Each message can be stored in a unique file name. An options menu is provided to provide message review, re-record and deletion.

Filespec format

The filename used for each message can be a unique name created by the Record control, and can also contain values from previous results in other controls. A wild char '*' can also be used, indicating a unique file name should be created. The Record control does this by replacing the * with random characters up to the maximum 8 character filename size. There should be no characters between the * and the . separator before the file extension. If a * is not specified, the same file will be overwritten on every record operation.

Options greeting

At the end of the recording, VBVoice can play a options greeting to allow the caller to choose options to re-record, delete, append to end of recording, or to save the message.

Database Access

The Record control can be associated with a data control to provide storage of received message files. The Record control selects the table to be updated using the RecordSource field in the dialog. Normally a table name is constructed from some fixed characters, and some digits collected to form a mailbox number. The Record control adds a new record to the database for each message taken. The table used must contain a field of name Filename, which is used to store the filenames of messages.
The data control used by a Record control can be shared with other Record controls, but not with DataFind or DataNew controls.

Database table name checking

The table name used for database update must be valid. An invalid table name can cause a Visual Basic message to pop up, stopping the channel until someone presses the OK button. The table name can be validated by using another database table.

Message File Database

The Record control can update a message file as used by the PlayMsgs control with the new message filename. This provides automatic access of received messages by the PlayMsgs control. This allows messages associated with a data control to provide storage of received message files. The Record control selects the table to be updated using the FilePath field in the dialog. The Record control adds a new entry to the end of the message file.

Error handling

If an invalid option digit is received, and code in the Invalid event does not process the digit, VBVoice will increment the error count, play the error prompt and the options greeting again.
If no speech was detected during the record operation,VBVoice will increment the error count, play the silence prompt and initial greeting, and restart the record operation.
If the error count exceeds the maximum error count, the call is transferred to the control connected to the Error node. If no control is connected, the system plays the default error message and terminates the call.
If no digits are detected after the options prompt, the message is automatically saved and the call will exit via node 0 to the next control.

Caller hangup

A database record is created containing the filename if:

hangup occurs after the recording
the caller does not press any digits
the caller presses the configured digit for save-message.

Filename

Applies to Record

Array String (1, MAXCHANNELS)

This property is the filename created from the File Spec in the setup dialog, and is the file that will be recorded. Your VB code can override the filename by setting the property in the Enter event. Changing the filename at any other time will have no effect. The filename can also be used as part of a greeting in order to replay the message. If the message is canceled or not recorded, the filename property will be a null string.

Example 1

To access the filename:

```
mystring = Record1.Filename(channel)
```

Example 2

To override the filename:

```
Record1.Filename(channel) = myfile.tmp
```

MaxSil

Applies to Record

Array Integer(1, MAXCHANNELS)
This is the duration (in seconds) of silence before VBVoice decides that the caller has stopped talking. This value is set by the control to the value Maximum Silence set in the setup dialog. It can be changed in the Enter event to another value if required. The new value only affects the current call.

MaxTime

Applies to Record

Array Integer(1, MAXCHANNELS)

This is the maximum duration (in seconds) allowed for the recorded message. This value is set by the control to the value Maximum Silence set in the setup dialog. It can be changed in the Enter event to another value if required. The new value only affects the current call. a

Example

To override the maximum time setting:

```
If CallerName = Fred Then
        Record1.MaxTime(channel) = 5  stop after 5 seconds silence
                                      Fred is a slow talker
Endif
```

Prompts used

The Record control uses some special prompts stored in the file RECORD.VAP. These prompts are indexed by position in the file: if they are changed, the file order must remain the same.

Main greeting

This will normally say "Please leave a message at the tone. Press # key or wait for further options, or simply hang up", depending on options set. The default greeting is Please leave a message at the tone.

Prompt after record.

This will normally say "Press 1 to save your message, 2 to re-record your message, 3 to delete your message, or 4 to save your message". This greeting should be changed to match the digit options you have set.

Prompt after error.

This prompt will inform the caller that they have pressed an incorrect digit. The prompt after record will then be played again. The default Error greeting is that was not a valid entry.

Prompt after silence.

If no voice is recorded after the initial greeting, this prompt is played. It might say "I'm sorry, I did not hear your message". The initial greeting is then played again and another attempt at recording is made, up until the error count is exceeded. There is no default Silence Timeout greeting.

Record dialog

See also Record

Record Start

Start after digit
Not implemented!

Filename
Enter a file specification here. Record will create a file based on this specification. See *Filename Format.*

Compress Silence
If checked, the voice card will remove all silence periods greater than a predefined size. This is set to a default size of 200ms, and can be changed using a setting in the CPL file.

Add control
This is a list of controls that have default properties. Controls on all displayed forms are listed. Use Cut and Paste to insert the name into the text field, or use the Add control button.

Record Termination

Maximum time for record
This fields sets the maximum time for a recording, in seconds.

Max silence
This field sets the maximum period of silence before terminating a record.

Minimum time for valid recording
A recording less than this time will not be regarded as valid and will be ignored. (Presently fixed at zero)

Update Database

Use data control
Check this button if you want the Record control to create a new database record and add the new message into the Filename field.

Database
Select a data control on the form to be used to access a database.

RecordSource
This field specifies the names of the mailbox tables. Record will use the RecordSource field to set the RecordSource property in the VB data control so that it updates the correct mailbox table for each caller.

Use message file
Check this button if you want Record to add an entry to a message file. Use this option if you are accessing the messages using a PlayMsgs control.

Source Mailbox
This optional field specifies the source mailbox for this message. If caller-id is implemented, this field can refer to a control which has translated the caller-id to a mailbox number. This number is saved with the message for use when replaying the message.

Destination Mailbox
This field specifies which mailbox file to update with the filename of the recorded message, when Use Message file is set.

Options after record

Options Prompt
Use this button to create the greeting that tells the caller what message options are available after playing each message.

Rerecord message

The record process is restarted, with no prompt.

Cancel
The message message canceled is played, and the call exits out of the NoMsgs output.

Review
The recorded message is played back to the caller. The options prompt is then played again.

Append
Not implemented at this time. Call for availability.

Terminate and Save
This digit will terminate the record process, save the file and exit the control via the Done node.

Error Handling

Invalid digit prompt

No digits prompt
These two buttons allow you to create a custom greeting to be played after invalid input from the caller has been received, or a valid recording was not created.
If the caller enters invalid digits after the options greeting, the Invalid event is called. If the Invalid event does not process the digit, the Invalid digit prompt is played, followed by the options greeting, and Record starts collecting option digits again.
If the caller does not leave a message, the silence prompt is played, followed by the options greeting, and the record operation is restarted.

Properties

- Decision
- GotoControl
- GotoNode

Events

- Enter
- Exit
- VoiceError

TimeSw control

Properties Events
Default Property: Decision

TimeSw Dialog
TimeSet Dialog

Description:

Transfers the call to another control according to time of day and day of week. Several timezones can be set for each control. If the current time and day of week do not match any of the fields specified, the call continues via the default connection.

Decision

Applies to TimeSw

Array String(1, MAXCHANNELS)

The Decision property is the name of the node by which the call leaves the control. This can be used if you wish to select subsequent greetings based on time, but otherwise perform the same operations independent of time. See also the System Phrase phrase type Initial Greeting.

TimeSw dialog

See also TimeSw

The TimeSw setup dialog shows the current list of valid times for this control. Use the Delete, New and Change buttons to change this list, or double click on an entry to change an entry.

Time Set dialog

See also TimeSw, TimeSw Dialog

The Time Set dialog allows you to create new and modify existing time specifications. Each entry specifies a time range in one day. In addition, you can select which days of the week will be valid for this time. Times are specified using 24 hour time.
Use the Add and Remove buttons to move days of the week to and from the *Include days* and *Available days* list.

Properties

- GotoControl
- GotoNode
- Value

Events

- Enter
- EnterB
- State01 - State10
- PhraseError
- PlayRequest
- VoiceEvent
- VoiceError

State01 - State10

Applies to User

When an event is received from the voice card driver, either a VoiceEvent or a State<xx>. Event is generated. If the state machine variable is 0, a VoiceEvent will occur. If the state variable is between 1 and 10, a State<xx> event will occur.

VoiceEvent

Applies to User

If there was no greeting set, a VoiceEvent event occurs immediately after the Enter event, with the eventcode set to EVT_EOF. If there was a greeting set, the VoiceEvent occurs after the play greeting terminates. It can terminate either due to a digit being received (EVT_MAXDT), or due to the end of file (EVT_EOF).
Since the User control itself does nothing except provide the events, our code must either call another voice card driver function to generate another event, or it must transfer the call to another control using the GotoNode or GotoControl properties.
Use the function vbv_setstate to set the state machine variable to a new value. The state machine is initially set to 0 when a call enters the User control. The state variable defines which event procedure will get called on the next voice card driver event.

WARNING!
If no voice card multi-tasking function is called, and the call is not transferred to another control, the channel will hang in the User control. This property of the User control can be useful when say conferencing a call when you want the caller to wait. When ready to restart the call, generate an event in the control by calling vbv_queue_event for the correct channel. This function can be called at any time if the channel is idle. In the event procedure, you can set the GotoNode or GotoControl properties to start processing of the call again.

User control

Properties Events
Default Property: Value

Description:

This control has no predetermined action. It allows Visual Basic to define all operations and provides complete control over the voice card. Examine the project in the USERDEMO directory and the description below to see an example of how to use this control.

Events

Enter and EnterB events are generated when control passes to the main input or input B. VB Code can call any driver function, and is responsible for transferring the call to another control when required using the GotoNode and GotoControl properties.

There is no Disconnect event in this control: the Voice event must detect this condition and hang up the call if required.

State Machine support

The User control has built-in support for state machines. Each User control has 10 built in event functions which can be used to create a state machine. To use the state machine, call the vbv_setstate function in the VoiceEvent event, with the state parameter set to 1, and then call a voice function. When the function completes, the State01 event will occur. You can now call another voice function, and use vbv_setstate to set the state to 2. When this is complete, the State02 event will occur - and so on. If you run out of states, you can transfer to another control, or alternatively create your own state variable, and use the Select statement to branch depending on the state. See Events below for more information.

UserDemo example

In this example, the VoiceEvent code opens a file and plays it using the Dialogic function *xplayf*.
If there is an error while opening or playing the file, the call is routed to Node 3 using the GotoNode property. Otherwise, we set the state machine to state 1 using the vbv_setstate function, and exit the event procedure. The file will now be played.
The next event from the Dialogic driver will cause the State01 event to occur. In the State01 procedure, the call is transferred to the control connected to node 0 using the GotoNode property.
When the play completes, a State01 event will occur.

Error handling

There are no predefined errors in this control. VB code must handle all events, and can invoke the default error handler, silence handler or caller hangup handler using VBVoice functions.

Why a state machine?

State machines are often required in voice processing systems since most voice functions do not wait until the operation is complete before returning to the calling program. The program must wait until it has received an event indicating that the operation is complete before continuing. A state machine is a way for the program to remember what it was doing between events, and to handle multiple channels at the same time.

Value

Applies to User

Array String(1, MAXCHANNELS)

The Value property can be set by code, and used subsequently in other controls to configure greetings and in other fields that accept Control result values.

Example
To set the User value property:
User1.Value(channel) = this is the result

GotoNode

Applies to All controls

Array Integer(1, MAXCHANNELS)

This property can be set in most controls by Enter event code.
Setting this property to a valid node number in the control where the event occurs will route the call to the control connected to that node. Node numbers start at 0, numbered from the top.
For instance, if GotoNode is set to 0, the call is passed to the control connected to the first (top) node.
The normal action of the control will not be performed.
An Error event will be generated if the node number is invalid, or not connected. Transfer takes place as soon as the Enter event code is completed.

See also GotoControl

Example
To set the next node:
GetDigits1.GotoNode(channel) = 4

GotoControl

Applies to All controls

Array String(1, MAXCHANNELS)

This property can be set in most controls by Enter event code. Setting this property to a valid control name will route the call to that control. The destination control does not need to be connected to the current control, and does not need to be on the same form. If GotoControl is set to "MailBox", the call is passed to the control named MailBox. The normal action of the control will not be performed. If the destination control is part of an array of controls, the control name used should be of the form ControlName(index), e.g. GetDigits(23).
An Error event will be generated if the node name is invalid. Transfer takes place as soon as the Enter event code is completed.
See also GotoNode

Example
To set the next control:
GetDigits1.GotoControl(channel) = User1(5) go to #5 in the array of User1 controls

Event handlers

System Response

When Visual Basic code is executing in event procedures, VBVoice will be prevented from responding to the caller until the code has finished executing. Most VBVoice events occur when the caller is waiting for a response from the system, so system response is important. If you wish to initiate a lengthy activity from a VBVoice event, use a Delay control, and put the code in a DelayStarted event handler. If the delay time is set to 0, the caller will be put on hold until your code has set the DelayTime property again. See *Yielding* below.

Yielding

Code that does not yield to the Windows scheduler can also prevent VBVoice from receiving voice events, so it is important that lengthy Visual Basic operations allow the rest of the system to run. This can be achieved by placing the Visual Basic DoEvents() function or DoEvents statement at frequent intervals in your code. Note that if calling DoEvents, your event procedure may be entered again by another channel while in DoEvents. All variables should be arrays indexed by the channel number to ensure that they are not overwritten by another channel (unless you desire this behavior). You should not call DoEvents from any event except DelayStarted unless you wish to stop voice processing on the channel in question.

VoiceError Event

Applies to All controls

This event occurs if a non-fatal error occurs in the control. A non-fatal error is one where an unexpected situation has occurred and the control does not know how to deal with it, but does not mean a hardware malfunction. Normally the call will be hung up if the error is not processed. This event allows VB code to intercept the error, analyze the situation and decide which control, if any, should continue processing the call. When an error event occurs, the control generating the event is not able to continue normal processing. Your code can choose to ignore the error and redirect the call to a new control using the GotoNode or GotoControl properties.
Otherwise the call will be terminated.

Sub VoiceError(Channel as Integer, ErrorType as Integer, ErrorData as Integer)

Examples of errors include invalid filenames in greetings and record filenames.

Error types:	**Data**
ININOTFOUND	-
CALLABANDON	-
DATABASE_ERR	-
UNEXPECTED_EVENT	The control received an unexpected event from the voice card driver.

PhraseError Event

Applies to all controls that have greetings.

Phrase errors can occur when VBVoice is unable to open a specified file, or cannot find a specified phrase in a file, or an invalid parameter was specified in a System phrase. Most errors are found either during design checking or at startup, but some errors occur at run-time for system phrases that contain calculated values, or if files are erased after startup.
In these cases a PhraseError event is generated in the control that is playing the phrase..
Your code can do one of the following:

- replace the phrase in error with a new phrase
- ignore the phrase
- allow the default action to occur - which will terminate the call.

Function prototype:

Sub PhraseError(Channel as Integer, Phrase as Long, ErrType as Integer)

To ignore the phrase:
Set the Phrase parameter to zero.
To insert a different phrase in its place:
Create a phrase using the vbv_create_phrase functions, and setting the Phrase parameter to this phrase.
To avoid infinite loops, if an error occurs while processing the replacement phrase, an event is not generated.

Possible ErrTypes:
GLOBAL CONST IDX_FILEOPENERROR = 0
GLOBAL CONST IDX_VAPPHRASEIDNOTFOUND = 1
GLOBAL CONST IDX_BADCONTROLNAME = 2
GLOBAL CONST IDX_BADTIMESPEC = 3
GLOBAL CONST IDX_BADNUMBER = 4
GLOBAL CONST IDX_BADDATESPEC = 5
GLOBAL CONST IDX_VAPPHRASENAMENOTFOUND = 6

PlayRequest Event

Applies to all controls that have greetings.

This event occurs when a greeting contains a VBPhrase. Each VB phrase is named so that it can be identified. VB code should create a phrase object using the DLL functions
.
return a phrase object in the PhraseObject parameter, or 0 to skip the phrase.

Sub PlayRequest(Channel as Integer, PhraseName as String, PhraseObject as Long)

Enter, EnterB Event

Applies to All controls (except Phone)

Either the Enter event or EnterB event will occur when a call enters a control, depending on the input by which the call arrives. The EnterB event can occur in controls that have 2 input nodes, when the call enters via the secondary input.

Enter events give VB code an opportunity to change some of the characteristics of the control by setting its properties on a call-by-call basis, or to bypass the control altogether by setting the GotoNode or GotoControl properties.

The Enter event has an additional parameter, Greeting. This parameter can be set to a greeting object, created with one of the DLL greeting functions. This allows your code to create greetings composed at runtime. This allows more flexibility in composition of greetings. See Custom Greetings .

Sub xx_Enter(Channel as Integer, Greeting as Long)
Sub xx_EnterB(Channel as Integer)

Exit Event

The Exit event occurs when a call leaves a control and is transferred to another control.

This event can be used to set variables after a voice action or subsystem has completed, or perform other operations.

Sub Greeting1_Exit (Channel As Integer, Node As Integer)

The node paramater indicates the node by which the call is leaving the control. Nodes are numbered from 0.

Applies to All controls except Onhook and User

Disconnect Event

Applies to All controls

This event procedure is called when a caller hangs up. It can occur in any control that does voice processing. This event allows clean-up on a per-control basis. After this event occurs, a global Disconnect event occurs in the Phone control that initiated this call. This hierarchy of events allows localized clean-up procedures and/or global clean-up as appropriate. The global Disconnect event occurs regardless of whether the caller hangs up first or the system hangs up the call. The local Disconnect event only occurs when the caller hangs up. If the reason for hangup is either CallerHangup or Default error handling, the NoHangup parameter can be set to TRUE to prevent the system from going onhook.
There are three possible reasons for hangup: caller hangup detected, default error handling, or the call arriving at an Onhook control

Sub Disconnect(Channel as Integer, Reason as Integer, NoHangup as Integer)

Possible disconnect reasons:

Const CONTROLHANGUP = 0	the call terminated at an Onhook control
Const SYSERRORHANGUP = 1	the call terminated due to a system error in a control
Const CALLERHANGUP = 2	the call terminated due to the caller hang-up indication from the telephone switch
Const INVALIDHANGUP = 3	the call terminated due to default invalid digits or silence timeout handling

VoiceCard Options dialog

Hookswitch options

Go offhook at startup
This option is useful if you are using a Skutch box or other device that gives you instant connection to the voice card having to dial. As soon as VBVoice starts, it will go offhook and establish a connection immediately.

Wait for ring
If you are using a PBX or line simulator, in which you have to dial a number to establish a connection, select this option. When VBVoice needs to use the line, it will wait for ringing before going offhook and continuing.

Stay offhook after test
If you had to dial in to make the connection, this option allows you to keep the connection alive, removing the need to dial in again for the next test or greeting.

If you lose the connection...
If you do hang up during a call, you can make the voice card go onhook using the offhook / onhook buttons in this dialog and also in the test dialog. You can then dial in again.

Choose Phrases dialog

See also: Add VAP Phrase dialog , Voice File dialog , System Phrases

This dialog is where you create your greetings. The Phrase List is a list of VBVoice phrases, which can be VOX files, VAP phrases, or System Phrases.

Play List button
This command will play the entire greeting. This allows you to check the concatenation of the different phrases. If your greeting contains VBPhrases, it is likely that references will be made to control properties. For each reference to a control property, VBVoice will show a dialog asking for the value for this property.

Play Phrase button
This command will play the currently selected phrase.

Remove button
This command will delete the currently selected phrase from the greeting.

Add System Phrase.
See System Phrases

Add VAP Phrase
See : Add VAP Phrase dialog

Add File
See Voice File dialog

Add VAP Phrase dialog

The Add VAP Phrase dialog allows you to add a phrase from a VAP phrase file to the greeting. The phrase files must be located in the default VBVoice directory. Once you have found the phrase you want, double click on the phrase or press OK to insert the phrase into the list.

VAP file list
This list box shows all the VAP files currently available. When you select a new file, the phrase list updates to show all the phrases in the list.

Phrase list
This list box shows the phrases in the current VAP file.
When you select a new file, the phrase list updates to show all the phrases in the list.

Show phrases containing
The phrases shown may be filtered by the Show phrases containing field. Whenever this field is changed, the phrase list is updated to only show the phrases containing the text shown. If the field is blank, all phrases are shown.

Find phrase
This command searches in the current file for a particular word or phrase. Searches are case-insensitive.

Play phrase
Plays the currently selected phrase.

Edit phrase
Starts or switches to the Announce! voice editor, and opens the selected phrase. Changes made in Announce! will not be reflected in this dialog until the file is saved.

New phrase
Use this command to create a new, empty phrase in the phrase file. Enter the required script for the phrase, which is then added to the phrase list. Use the Edit phrase button to add the voice for this phrase. Once in the voice editor, you can create more new phrases. These will be added the list once you save the file in Announce!, and return to Visual Basic.

System Phrases

The System Phrase dialog is used to add a System phrase to a greeting. System phrases are defined to say money, times, dates, numbers and other system phrases. Most system phrases require a parameter. Use the Find Control button to select a control name and add it to the parameter field. At run time the actual value of the controls default property will be substituted for the control name in the parameter string.

Date Parameter: a 4 or 6 character string. The first 4 characters are mmdd or ddmm depending on date format selected for the current language. The last 2 characters are an optional year field. See Date and number formats

Digits VBVoice says a number or string as a series of digits and letters

FileSpec Parameter: the name of a voice file in the VBVoice directory, or a full path to a voice file. Example: if the parameter field is BOX%MAILBOX%.VOX, and the MailBox control has collected the digits 123, VBVoice will attempt to play a file BOX123.VOX from the default VBVoice directory.

FileDate As above, but VBVoice says the last modified date of the file.

FileSize Parameter: a file path. VBVoice will say the size of the file. The path can contain control names. The file path should be a fully qualified path name.

FileTime As above, but VBVoice says the last modified time of the file.

Initial Greeting This phrase is intended to be used as the first phrase in the main voice system prompt. It says Good Morning, Good Afternoon, or Good Evening based on time of day. Before noon, it says good morning, from noon to 5 PM, it says good afternoon, and from 5 PM to midnight it says good evening. The phrases used are in VBASE.VAP and can be changed to suit your requirements. The position of the phrases in the file should not be changed.

Money VBVoice translates a number into an amount, e.g. 1234 is "twelve dollars and 34 cents". The dollars and cents phrases are in VBVOICE.VAP and can be replaced with your own units.

Number VBVoice says the numeric value of the parameter, i.e. the string 1563 would be 'one thousand, five hundred and sixty three'. This phrase type can also handle decimal numbers, e.g. 123.45 or 123*45 would say one hundred and twenty three point four five. See Date and number formats

Number (Ordinal As Number, but says first, second etc. Decimal points are not allowed in this number.

NumberShort As Number, but numbers greater than 9 are said as "more than nine".

NumFiles Parameter: a file path that contains a drive, directory and wild cards i.e. MAILBOX*.vox. VBVoice will say the number of files found that match the file specification.

Rotator Parameter: VAP file name. A rotator is a phrase that says one of a number of phrases in rotation. The phrase to be played is selected from the VAP file in sequence, regardless of the channel upon which the phrase is played.This is useful when you are playing advertising messages, and want to give each phrase an equal amount of exposure.

Time Parameter: a time in the format hh:mm.

VAP Phrase by Index Parameters: a VAP file name, and an index number. The index number selects the phrase to play. Phrases in the VAP file are numbered from 0;

VAP Phrase by Name Parameters: a VAP file name, and a phrase name (script). The phrase name can contain control names to allow phrases to be selected at runtime by control properties.

VBPhrase This is a phrase type that allows VB code to create a phrase at runtime. A name is associated with the phrase so that each phrase can be identified. When a control attempts to play a greeting containing a VBPhrase, a PlayRequest event will be generated. VB code can then set the file or phrase to play. There can be as many VBPhrases as required in a greeting. Phrases can be created using the DLL functions. There is no limit on the number of phrases in a greeting.

A VoiceError event (type PHRASE_ERROR) is generated if an error occurs during phrase processing, such as the specified file does not exist, or an invalid Control name has been specified. See Errors

Voice File dialog

The file dialog allows you to add a compressed ADPCM voice file to play in your greeting. If you wish to dynamically select a file to play based on previous digits or database entries, use the System phrase type File Spec instead. The File dialog is a standard Windows file selection dialog, except VBVoice adds a button Play. This will play the selected file before you add it to the greeting.

Event Log Options dialog

The Event Log Options dialog is used to control which events are shown in each channel log window, and also to control which events are logged to the log file.
Channel log windows will display events as they occur, and also can extract events from the log file and show them.

Note that having many log windows open may slow down your system.

Event types to monitor:

errors	
calls	the start and stop time for each call,and the system start and stop times
call flow	a call flow message is logged whenever a call leaves or enters a control
information messages	these are general information messages about the current action of each control
invalid digits, time-outs	these log the events that occur when a control detects a timeout or invalid digits from the caller.
driver events	low level voice driver events
driver functions	low level voice driver function calls
control states	These messages reflect the internal state machines used by each control

The VBVoice Log File

A new log file is created every day to avoid the log file becoming unnecessarily large. The file name for the log files are of the form vbvdd-mm.vlg, where dd is the day, and mm the month. Log file size may still become a problem if all events are monitored to the log file. Normally calls, call flow and error events should only be logged to the log file. You can choose which event types to log using the LogFiles item in the Setup menu.

VBVoice Main Window

Main Menu

Check Menu

Use the Check Control , and Check System commands to check your system. Even if you have only one form in your system, Check System performs some extra checks over Check Form. If you get errors, double click on the error message to highlight the control in error

Check Control
Check control will perform an error check on the currently selected control. If a log window is not visible, one will be created to display the results of the check.
Check Form
This command will perform a Check Control command on all the controls on the current form.
Check System
This command will perform a Check Form on all forms currently loaded. Note that forms in the project that are not displayed, either normal size or minimized, will not be included in the check.

Test Menu

This command will start VBVoice in Test Mode. The test will not start until you use the Start Test command. See Test Mode, page **Error! Bookmark not defined.** in the next chapter.

Sound Menu

Use Voice card
Use Sound card
These commands allow you to choose which sound device to use for listening to greetings, and for testing the system.
Voicecard Options
This command shows the Voice Card options dialog. This dialog allows you to configure how you will be using your voice card during testing. See the Voice Card Options dialog.

Window Menu

Line Status
This command is enabled when VBVoice has started the voice system. It will display the Line Status window, if not displayed
Log Windows
This command allows you to select which channels you want to monitor as your system is running. The dialog shows which channels are currently being monitored with a check mark in the box for that channel. When you click OK in the dialog, log windows are opened or closed to reflect the status you entered in the dialog. Showing too many log windows may slow down your system especially when monitoring all events. See Log Windows, page 122.

Setup Menu

Directories
This dialog allows you to change the path to ANNOUNCE.EXE, the voice editor, and the default directory that VBVoice uses to find phrase files.
PBX
This dialog allows you to change the way that VBVoice interacts with the phone system to perform operations such as transfer, put on hold, etc. These parameters are used in the Dial and Delay controls.
Error Handlers
Use this dialog to define the greeting files that will be played when the default invalid digit or time-out handlers are invoked. See GetDigits, PlayMsgs, Record and Count for details on when these error handlers are used.
Startup Files
This dialog allows you to specify which voice and phrase files will be opened when the voice system is started. Opening files at start-up time speeds up execution by eliminating the time required to open and close the file for each play operation. However, file handles are a limited resource in the system, so files

that are only used occasionally should be opened when required.
Before using this dialog, load all your voice forms so that VBVoice can scan them all to produce an accurate list of all voice files in use.

System Menu

Start System

When Visual Basic is in run mode, the Start System menu item is enabled. Start System performs the same function as the DLL function vbv_start_system(). It checks all the controls, loads the voice card driver and start voice operations. It also displays the Line Status window, unlike the vbv_start_system function.

Stop System

Once the system is started, you can stop voice operations with Stop System. Normally VBVoice will wait for all lines to clear before terminating. The following events will occur:

- The Line Status window will appear and will mark all idle lines as waiting for termination.
- It will then show the Wait for Termination dialog while waiting for lines to clear.

If you want to terminate immediately, use the Stop All Lines button in the dialog. All lines will be cleared immediately, and the system shut down. Otherwise VBVoice shut down the voice system when all lines have been released.

Forcing a system stop...

If no valid events are received from the voice card driver, it may not be able to complete an orderly shutdown. If this happens, close the dialog using the dialog System menu (use the button at the top left of the window) and select Close to force termination. Some greeting files may be left open, so the application should be restarted.

Log File Setup

This command will allow you to specify the level of detail for the log file. See Event Logs .

Test Mode

General Considerations using Test Mode

Testing Subsystems and Individual Controls

Test mode allows you to start a test of your system at any point in the call flow diagram. Show the Test Mode dialog by selecting the Test menu item in the status window. The control that will start the test is highlighted (white text on black) and is shown in the test dialog name. You can change the start control by clicking on the required control, or it can be selected via the Start Control**...** button.

Test mode is useful for testing a subsystem without having to worry about how the call is normally routed to the subsystem. In complex multi-level menus, it is convenient to be able to start at any point in the system, however, consideration must be given to the interactions between controls in your system. Consider the scenario where a Record control uses control name substitution from a previous GetDigits control to create a filename based on the entered mailbox digits (to direct the file to the correct mailbox directory). If test mode is started at the Record control, the GetDigits control will not have collected any digits. VBVoice detects this condition, and will present a dialog requesting that you fill in the value that the GetDigits control would have supplied. Again, you can enter an invalid value to check all error conditions.

Check System

It is not essential to use the Check Control, Check Form and Check System commands before starting test mode. VBVoice will check each control before it transfers the call to a new control. However, it will often save time to have an error-free design before testing.

Call Flow

You can see you how your call is progressing during the test. The call always starts in the control with the highlighted name. If the call moves to another control, the originating node and the line connecting the two are highlighted. If an error occurs, or the call is terminated with the Stop button, the control name of the last control is highlighted in dark red, allowing you to zero in on error locations quickly.

Timeouts and Caller hangup

The test dialog provides Timeout and Hangup buttons which allow you to simulate a time-out while waiting for digits, or a caller hangup at any time.

Outdialing and Call Progress

When in test mode, VBVoice will dial but will not perform transfers or call progress detection. It allows you the select the call progress condition that you want to test from a dialog, so you can test all control paths without having to recreate (and wait for) ring-no-answer, busy, etc..

Testing in Visual Basic design mode - limitations

Test mode can be invoked while Visual Basic is in design mode, and is often useful as a quick test of part of your design, however it does have limitations.

Database access:

Database access is not available in design mode mode. If the call arrives at a DataFind control or DataNew control in Test mode, and Visual Basic is in design mode, the test will terminate.

Visual Basic code:

If your design relies on Visual Basic code, in custom VB phrase events, Enter events event or elsewhere, this code will not be run while in design mode.

Testing in Visual Basic run mode

Test mode can be invoked while Visual Basic is in run mode. In this case, database access is available and Visual Basic event procedures will be called.

You must ensure that Visual Basic loads the required forms at startup. This is a requirement for normal operation as well as for test mode. If only one form is to be tested, it can be made the startup form in the project options dialog (in the Options menu). If more than one form is required, the form load procedure in the startup form or module must load all other required forms. You can load the form as an icon by setting

the WindowState property to 1.

Visual Basic debugging:

Test mode does not interact with Visual Basic debug mode, as it does when running using Start System (see Running your Application). If Visual Basic switches to break mode due to an error in your code, Test mode will terminate.

Using the Sound Card in Test mode

A sound card can be used for playing greetings in design mode and for testing your system. VBVoice simulates actions relating to the telephone system such as dialing.

Dialing

During dialing operations, VBVoice will say the number dialed using the sound card. If call supervision is being performed, VBVoice will show a dialog allowing you to select the call progress. Call progress Analysis type to be simulated - call answer, busy, no dialtone etc. This allows you to test the system without having to wait through 30 seconds of ringing to test the no-answer handling.

Dialed digits

Dialed digits can be entered at any time using the dial pad in the Test dialog as if you were the caller.

Recording

VBVoice uses prerecorded files as the source of data when making recordings. This saves time, and eliminates the need for a microphone. VBVoice shows a file select dialog when a record operation begins. The VOX file you select is copied into the file that would have been recorded.

Using the Voice Card in Test mode

Using the voice card in test mode operates exactly like a real call, except for the start and end of each call, and for outcalling.

The system will establish a call before the test starts using the configuration settings in the VoiceCard Options dialog. When the test is completed, the system will not automatically go back on hook, but will leave the call established, ready for the next test.

If a test is started in a Phone control , VBVoice uses a dialog to ask if the call should continue via the Ring or StartCall connections.

If a test using a Dial control, with supervision enabled, VBVoice will dial the digits, and then ask you via a dialog what the result of the call should be: answer, busy, etc.

VBVoice Test Dialog

When VBVoice test mode is started, access to the Visual Basic and VBVoice main menus is denied. Terminate test mode with the Cancel button to return to normal operation.

Start In ...

This command allows you to select the control at which you want to start the test. You can also do this by clicking in the control.

Start / Stop

Use this command to start the test. Once the test is active, you will not be able to make any design changes until the test is stopped.

Timeout

The time-out command allows you to simulate a time-out condition when performing a delay operation, or a silence time-out when performing a get digits operation. This allows you to quickly test time-outs without having to wait for the time-out to actually expire.

HungUp

The HungUp command simulates the caller hanging up the phone to terminate the call when using a sound card, and avoids the need to drop the line when using a voice card.

Pause/Resume (Sound card only)

When using a sound card, you can pause play, get digits and delay operations until you are ready to resume.

Cancel

This command will terminate test mode and will return the system to normal operation.

Dial Pad (Sound card only)

The dial pad is used to enter digits from the caller when testing using a sound card.

Onhook / Offhook (Voice card only)

The onhook/offhook buttons show the current status of the hookswitch when using a voice card. Normally when testing a voice system you will want to stay offhook and connected to the voice card for the duration of the test - this saves time dialing and waiting to be reconnected each time. However if you lose the connection, you can use the onhook button to clear the line and dial in to the voice card again to re-establish the connection.

DLL Functions

Status Window Functions

- vbv_show_linestatus
- vbv_show_log

Phrase Functions

- vbv_create_voxfile_phrase
- vbv_create_named_phrase
- vbv_create_phrase
- vbv_create_sys_phrase
- vbv_destroy_phrase

Greeting functions

- vbv_create_greeting
- vbv_add_phrase_to_greeting
- vbv_clear_greeting
- vbv_destroy_greeting

General Functions

- vbv_add_msg
- vbv_delete_msg
- vbv_get_voice_channels
- vbv_log_msg
- vbv_query_line
- vbv_set_directory
- vbv_set_language
- vbv_setstate
- vbv_start_system
- vbv_stop_system

Multiple language support

VBVoice can support up to 12 languages simultaneously if your system is authorized for the multi-language option.

Requirements for multi-language support

There are three basic requirements for multi-language support in a voice system

- each caller must be able to select their own language.
- the actual voice files to be played must be selected based on the current language.
- the system phrases for dates, times, numbers must be created using the format required for the chosen language.

The phrase files for the default language (language 0) are kept in the VBV directory. Additional phrase files for each language are kept in directories LANG1, LANG2 etc. up to the number of languages in use. These directories must be subdirectories of the VBV directory.

Date and number formats

Date and number formats

The formats used to say numbers and dates can be set for each language. These formats are controlled by VBVOICE.INI settings.

Numbers

There are two options when saying numbers between 0 and 99: either each number can be recorded separately, or the numbers can be created by concatenating the phrases 0 to 9, and 10 to 90 together. The default is to use concatenation. To use the first method, you must create 2 phrase files, VBV_NUM.VAP (for regular numbers) and VBV_ORD.VAP (for ordinals), each containing all the numbers from 20 to 99 in sequence. These files are available from PRONEXUS. In addition, set the Numbers entry in VBVOICE.INI to Full. This method provides better recording quality, and language independence at the expense of more disk space for the phrase files.

Dates can be said as

- mmddyy (name of month, day, year)
- ddmmyy (day, name of month, year)
- mdy (number of month, day, year)
- dmy (day, number of month, year)

The date format for the current language also specifies the way that dates are processed by the system phrase Say Date<>.

Example

This example sets language 0 to be English, language 1 to be French, and language 2 to be Spanish. Since French number creation does not follow normal English rules for some numbers (21,31, 70-80) we use the Full option for French numbers. In English we have set dates to be said using numerals only, month first (i.e 4 - 25 - 1994) whereas in French dates will be spoken in full, day first (April 25 - 1994).

```
[Languages]
Language0 = English
Language1 = French
Language2 = Spanish

[English]
Date= mdy
[French]
Date= mmddyy
Number=Full

[Spanish]
Date= mmddyy
Number=Full
```

Please note these important points:

- Each directory must contain a complete set of phrase files required for your application.
- Each language phrase file must have the phrases in the same order as the equivalent phrase file in the VBV directory.
- Only phrase files referenced without a path name will be substituted by the correct language file. If you reference a file directly using C:\VBV\VBVOICE.VAP, then this phrase file will be used regardless of the language.

To change languages

To change to another language for a particular call, use the DLL function *vbv_set_language*. This will not affect any other calls. Pass in the channel number that this call is operating on (available in the Enter and Exit events), and the language number. This can 0 (default language), up to 11.

Example:

vbv_set_language channel, 1

Each call starts with the default language, language 0. All voice files that do not have full path names will use the language setting to select the voice file location. If a phrase is created using a full path name, the language setting is ignored.

vbv_show_linestatus

This function changes the visibility of the line status window according to the show variable. If *show* is TRUE, the Line Status window is made visible, if it is FALSE the window is hidden.

Declare Sub vbv_show_linestatus Lib "VBVOICE.VBX" (ByVal show as Integer) as Integer

vbv_show_log

Changes the visibility of a VBVoice log window. If show is TRUE, the window is made visible. If show if FALSE, the window is hidden.

Declare Sub vbv_show_log Lib "VBVOICE.VBX" (ByVal channel as integer, ByVal show as integer)

vbv_create_voxfile_phrase

Opens the specified file and returns a handle to a new phrase object. When added to a playlist, the entire file is played. If an error occurs, 0 is returned.

Declare Function vbv_create_voxfile_phrase Lib "VBVOICE.VBX" (ByVal filename as String) as Long

vbv_create_named_phrase

Opens the phrase file, if not already open, and returns a handle to a new phrase object from the specified phrase. The phrase is identified by the phrase script which must match exactly the script in the phrase file. The phrase object must be deleted when finished with using vbv_destroy_phrase. Note that if a phrase is being played often, it is more efficient to find the index of the phrase using vbv_find_phrase_index, and then use vbv_create_phrase. If an error occurs, 0 is returned.

Declare Function vbv_create_named_phrase Lib "VBVOICE.VBX" (ByVal filename as String, ByVal phrasename as String) as Long

vbv_create_phrase

Opens the phrase file, if not already open, and returns a handle to a new phrase object from a phrase in a VAP phrase file. The phrase is identified by the index of the phrase in the file. If an error occurs, 0 is returned.

Declare Function vbv_create_phrase Lib "VBVOICE.VBX" (ByVal filename as String, ByVal phrase_id as Integer) as Long

vbv_create_sys_phrase

Creates a new system phrase object. Each system phrase takes a string containing the data required, as listed below. Note VBPhrases cannot be created with this function.

Declare Function vbv_create_sys_phrase Lib "VBVOICE.VBX" (ByVal type as Integer, ByVal parameter as String) as Long

The phrase type must be one of the phrase types as declared in VBVOICE.BAS. These correspond to the system phrases in the System Phrase dialog.

See System Phrases

vbv_destroy_phrase

Destroys a previously created phrase, and closes the file if necessary. Note that if a phrase is added to a greeting in a User control, the greeting object assumes ownership of the phrase object and will destroy the phrase.

Declare Sub vbv_destroy_phrase Lib "VBVOICE.VBX"(ByVal phraseid as long)

vbv_destroy_greeting

Destroys a greeting object, but does not destroy any phrases in the greeting

Declare Sub vbv_destroy_greeting Lib "VBVOICE.VBX" (ByVal greeting As Long)

vbv_start_system

Starts the VBVoice system. All VBVoice forms must be loaded prior to calling this function. Returns TRUE if successful, or FALSE if the system failed to start

Declare Function vbv_start_system Lib "VBVOICE.VBX" () as Integer

vbv_stop_system

Stops the VBVoice system. Show the VBVoice line status window, and if any lines are active, waits for call completion. The user can press the Stop all lines button to terminate all calls immediately. This function should not be called from a control event. If you want to trigger system shutdown from a VBVoice control event, use the disconnected event, or trigger a timer which can then call this function.

Declare Sub vbv_stop_system Lib "VBVOICE.VBX" ()

vbv_set_directory

Sets the default VBVoice directory. The path supplied should not have a terminating \ character.

Declare Sub vbv_set_directory Lib "VBVOICE.VBX" (ByVal pathname as String)

vbv_query_line

Returns the status of a line, including the name of the control currently handling this line, and an integer that represents the current state. This can be one of the error constants:
Declare Sub vbv_query_line Lib "VBVOICE.VBX" (channel as integer, controlname as string, state as integer)

The line status constants:

LINEIDLE = 0
LINEPLAYING_GREETING = 1
LINEWAITING_FOR_DIGITS = 2
LINEPLAYING_SIL_GREETING = 3
LINEPLAYING_ERR_GREETING = 4
LINEPLAYING_SENT_GREETING = 5
LINEPLAYING_CANCEL_GREETING = 6
LINERECORDING = 7
LINEPLAYING_DELAY_RECORDING = 8
LINEDIALLING = 9
LINEEVENTERROR = 10
LINEDELAY = 11
LINESTOPPED_ERROR = 12
LINESTOPPED = 13
LINECALLER_HANGUP = 14
LINESTOPPED_CALLER_HANGUP = 15
LINEPLAYING_OPT_GREETING = 16
LINECALL_COMPLETE = 17
LINEPRECALLDELAY = 18
LINEWAITRING = 19
LINESTARTINGCALL = 20
LINEINTERCALLDELAY = 21
LINEWAITSGHUTDOWN = 22
LINEWAITDIALTONE = 23
LINEWAITCHANNELSTOP = 24

vbv_setstate

This function is only used in the User control to set the state machine variable. See the User control, p **II-1**.

Declare Sub vbv_setstate(ByVal ch as Integer, ByVal state as Integer)

vbv_set_language

Use this command to set the language for a call. The language set will remain in effect for the duration of the call. When a language is set, a new set of phrase files is used (in directories LANG1 for language 1, LANG2 for language 2), and a new set of rules is used for system phrases. See Multiple Languages . The actual language for language 1, 2 etc must be specified in VBVOICE.INI under [Language]: Contact PRONEXUS for languages currently supported.
LANG1 = SPANISH
LANG2 = FRENCH etc.

Declare Sub vbv_set_language Lib "VBVOICE.VBX" (ByVal ch As Integer, ByVal lang As Integer)

(Must have language option installed)

vbv_create_greeting

This function creates a greeting object, to be returned in the Greeting parameter of the Enter event. The greeting object is initially empty: use vbv_add_phrase_to_greeting to add phrases to the greeting.

Declare Function vbv_create_greeting Lib "VBVOICE.VBX" () As Long

vbv_add_phrase_to_greeting

Tthis function adds a phrase to a greeting. The phrase must not be destroyed until the call has left the control. Phrase objects can be shared amongst different greetings and different channels, as long as they are not destroyed while still in use.

Declare Sub vbv_add_phrase_to_greeting Lib "VBVOICE.VBX" (ByVal greeting As Long, ByVal phrase As Long)

vbv_clear_greeting

Removes all phrases from a greeting object, but does not destroy any phrases in the greeting. Allows you to start an existing greeting object from scratch without having to destroy it and create it again.

Declare Sub vbv_clear_greeting Lib "VBVOICE.VBX" (ByVal greeting As Long)

vbv_destroy_greeting

Destroys a greeting object, but does not destroy any phrases in the greeting

Declare Sub vbv_destroy_greeting Lib "VBVOICE.VBX" (ByVal greeting As Long)

vbv_add_msg

This function is used to add a message to a mailbox. If your system needs to add a new message to a mailbox, use this function rather than modifying the mailbox file directly. See the Voicemail example, p I-1 for more information. Parameter channel is only used for logging purposes. The mailbox is the mailbox number e.g. 100 or 200 in the voicemail example as shipped.

Declare Function vbv_add_msg Lib "vbvoice.vbx" (ByVal recordedfilename as String, ByVal mailbox as String, ByVal channel as Integer, ByVal srcbox as Integer)

vbv_delete_msg

This function is used to delete a message to a mailbox. If your system needs to delete a message from a mailbox, use this function rather than modifying the mailbox file directly. See the Voicemail example, p I-1 for more information.

Declare Function vbv_delet_msg Lib "vbvoice.vbx" (ByVal recordedfilename as String, ByVal mailbox as String, ByVal channel as Integer)

vbv_get_voice_channels

Returns the number of voice channels found by the voice card driver. If the driver did not load, it returns 0.

Declare Function vbv_get_voice_channels Lib "VBVOICE.VBX" () as Integer

vbv_log_msg

Adds a new message to the VBVoice log file.

Declare Function vbv_log_msg Lib "vbvoice.vbx" (ByVal channel as Integer, ByVal message as String)

Custom Greeting example

In this example, the phrases and greetings are created in Form Load.
Note that phrase1 is added to the greeting twice.

```
mygreeting = vbv_create_greeting()
phrase1 = vbv_create_phrase("Voicmail.vap", 0)
phrase2 = vbv_create_sys_phrase(SayNumber, "123")
phrase3 = vbv_create_phrase("Voicmail.vap", 2)
If phrase1 = 0 Or phrase2 = 0 Or phrase3 = 0 Then
    MsgBox "cannot create phrase"
Else
    vbv_add_phrase_to_greeting mygreeting, phrase1
    vbv_add_phrase_to_greeting mygreeting, phrase2
    vbv_add_phrase_to_greeting mygreeting, phrase3
    vbv_add_phrase_to_greeting mygreeting, phrase1
End If
```

In the Enter event, the greeting is passed back to VBVoice to play

```
Sub Greeting3_Enter (channel As Integer, Greeting As Long)
Greeting = mygreeting
End Sub
```

Custom greetings

To create a single phrase that is configured at run time, use a VBPhrase. For more complex custom greetings, use this new custom greeting capability.
The consists of some new DLL functions, and a new parameter in the Enter event.

The functions are:
vbv_create_greeting
vbv_add_phrase_to_greeting
vbv_clear_greeting
vbv_destroy_greeting

vbv_create_greeting creates a greeting object that can be returned in the Greeting parameter in the Enter event. If you return a Greeting object, it overrides any greeting already set for that control.
You can create lists of phrases by adding them to a greeting object.
You can clear a greeting and start over by using vbv_clear_greeting. Note this does not destroy the phrases in the greeting.
If you are generating greetings specific to each call, then you should keep a global greeting object around, and in each Enter event, clear the greeting, add the phrases you want, and then return it.
When you destroy a greeting, the phrases are not destroyed - you must destroy these explicitly.
Phrases must not be destroyed until there is no possibility of them being played again - if they are added to a greeting in the Enter event, they must not be destroyed until the Exit event.
If they are not destroyed by your code, they will be destroyed automatically when the system stops.

Example

Using databases

Creating a database

You do not need a special application to create simple databases. Visual Basic provides a tool, DATAMGR.EXE which can create and update databases. While DataMgr is limited, it does provide a quick and simple way to view and create databases.

Setting up the Microsoft data control.

At first sight the Visual Basic data control appears to have a daunting array of properties, but it requires only 2 properties to be set in order to connect it to a database. These are DatabaseName and RecordSource.

The DatabaseName property defines the file containing the database. This file should be the Microsoft Access database file, created with Access or with the DataMgr utility. DATAMGR.EXE is located in your VB directory. If using Access, you can also set up attached databases, so that VBVoice can access remote databases from a database server.

The RecordSource property defines the recordset within that database that the data control will access. Although it is possible to use SQL statements in the RecordSource property, for most applications a table name will suffice.
If you are not writing to your database, you can improve performance by setting the ReadOnly property to TRUE.
To gain more experience with the data control, look at the Visual Basic sample program in the BIBLIO directory.

A sample SQL statement

```
SELECT Titles.Title.Dept, Author FROM Titles, Authors WHERE Titles.AU_ID =
Authors.AU_ID
```

When you set the RecordSource to a table name, the data control will be able to access all the records in that table. If you will not be updating the database, you can set the ReadOnly property to True.

Accessing the data

You can add some text boxes to your form, and link them to fields in the database. The text boxes will display the contents of the current record in the database, and will also allow you to change the data in the database.

To link a text box to a database, perform the following operations:

- Add a text box to your form. You can use a label also, but this will not allow you to change the data.
- Set the DataSource property to the name of the data control to which you want to link.
- Set the DataField property to the field name in the table.

Now, when you start your program running, the text box will show the contents of the first record in the database.
To scan through the database, use the arrows on the data control.

Changing the data

To change the database contents, put some new text in the text box. The database will be updated when you move the data control to a new record.

Overriding the DataFind search algorithm

The standard database search algorithm provided by DataFind can be slow in large databases, and does not support SQL searches that are defined at runtime. You can select the RecordSet at startup using the RecordSource property, however this should not be changed at runtime (unless you are sure that there are no channels currently using the data control). You can override the standard search algorithm to improve performance or to add SQL search statements based on control properties such as collected digits. (See below)

Improving database performance

The DataFind control searches the database using a MoveNext / GetData / Compare algorithm. This can be inefficient when searching large databases. To overcome this, override the standard search algorithm using the code described below.

This code overrides the normal action of the DataFind control to use the commands FindFirst and FindNext, while still supporting the other data controls.

In addition, the file PERFORM.TXT in your VB directory contains information relevant to improving database speed from within Visual Basic. In particular, you can improve performance by setting the database ReadOnly using the ReadOnly property of the data control. You can also increase the data control cache size with the maxbuffersize entry in VB.INI.

Also note that having text controls bound to the data control slows down searches considerably. These have been added to the VMDEMO and INVDEMO forms to show you the current contents of the database only. If you wish to display the current contents of the database on one of your forms, use another data control, or turn off the link from the text control to the data control during the search.

Using SQL statements as search criteria

If you wish to add more complex search criteria to your database searches, or use a different search mechansim than that provided by the DataFind control, you can override the standard search mechanism to incorporate SQL statements, as described below. The string MyCriteria in the example below can be any SQL string, and can include your own variables and VBVoice control properties.

Example

In the Enter event, add the code below. This code sets up a search criteria string, and uses the FindFirst method to find the first record. If a record is found, the GotoControl property is used to transfer the call to the control connected to the Found output, otherwise the call is transferred to the Not Found output. The position of the record found is stored in the property Bookmark so that the other data controls can access the correct record when accessing the datafields. Note error handling has been omitted.

If using the Use Global Bookmarks option, bookmarks would be defined as a single string rather than an array of strings.

```
Sub DataFind1_Enter (Channel As Integer, Greeting as Long)
        Dim MyCriteria As String

        move to the beginning of the recordset
        DataControl.Recordset.MoveFirst

        set search criteria for Student_ID database field to equal digits received
        MyCriteria = " StockNumber = '" + GetDigits1.Digits(channel) + "'"

        'do the find operation
        DataControl.Recordset.FindFirst MyCriteria

        ' remember the position for the EnterB event
        DataFind1.Bookmark(channel) = DataControl.Recordset.Bookmark

         this duplicates the found / not found branching
         setting the gotonode makes the call exit immediately, overriding the
         normal action of the datafind
        If DataControl.Recordset.NoMatch Then
        DataFind1.GotoNode(channel) = 2
        Else
        DataFind1.GotoNode(channel) = 0
        End If
        End Sub
```

The EnterB event (below) is similar, but uses FindNext instead of FindFirst, and if no match is found, it transfers to the Not Found exit, rather than the None Found exit.

```
Sub DataFind1_EnterB (channel As Integer)
        Dim mycrit As String

        goto bookmark for this channel
        DataControl.Recordset.Bookmark = DataFind1.Bookmark(channel)

        set search criteria for Student_ID database field to equal digits received
        mycrit = " StockNumber = '" + GetDigits1.Digits(channel) + "'"
        DataControl.Recordset.FindNext mycrit

        If DataControl.Recordset.NoMatch Then
        remember new bookmark
        DataFind1.Bookmark(channel) = DataControl.Recordset.Bookmark
        goto Found output
        DataFind1.GotoNode(channel) = 1
        Else
        DataFind1.GotoNode(channel) = 0
        End If
        End Sub
```

VBVOICE.INI File Settings

SECTIONS

Settings

[VoiceCard]

PIKA cards only

INI SETTING	DESCRIPTION	OPTIONS / UNITS	DEFAULT
TSR_Interrupt	Software interrupt setting for the Pikatsr driver		
Windows_Interrupt	Software interrupt setting for the Pika Windows DLL		

Loop_back_cp Call progress parameter

DIALOGIC/RHETOREX/NEWVOICE cards only

INI SETTING	DESCRIPTION	OPTIONS / UNITS	DEFAULT
Playrate	Voice file sampling rate Hz	6000	

BICOM cards only

INI SETTING	DESCRIPTION	OPTIONS / UNITS	DEFAULT
HW_Interrupt	Hardware interrupt setting for your voice card		
Playrate	Voice file sampling rate	Hz - 6000/8000	6000
MemoryBase	Starting address of card(s)		D000
DriverParams	Parameter string for BICOM DLL		

ALL CARDS

INI SETTING	DESCRIPTION	OPTIONS / UNITS	DEFAULT
DisableLCDetect	Controls loop current detection	1 = no detect 0 = detect	0
HideMonitor	hides monitor program	1 = hide 0 = show	0
Flash_character	the character to be used in dial strings for a hook flash		!
Flashtime	duration of a hookflash	ms	500
HideMonitor	hides VBV Monitor program	0 to show 1 to hide	0
LoadDriver	Controls loading of the voice driver	0 = dont load 1 = always load	1
MinLCOFFTime	minimum time for loop drop	ms	set by voice card driver
MsgBoxErrors	show message boxes from DLL on startup failure if = 1	1	
Offhook_delay	adjusts offhook delay default	ms	500
PauseTime	duration for a pause character (comma) during dialing	ms	200
Ring_cnt_reset	time after which the driver resets the ring count default set by voice card driver	ms	set by voice card driver
SkipBBytes	sets the number of bytes to skip backward during rewind in the greeting control (not playmsgs)	ms	8000
SkipFBytes	sets the number of bytes to skip forward in the voice file on fast forward in the greeting control (not playmsgs)	ms	4000
Type	Type of voice card in the system		NONE

[Record]

INI SETTING	DESCRIPTION	OPTIONS / UNITS	DEFAULT
RecChop	deletes bytes from end of recorded messages for deleting tones	bytes	0
RecChopSil	deletes silience from end of recorded messages when ending in silence	0 / 1	0

[Phone]

INI SETTING	DESCRIPTION	OPTIONS / UNITS	DEFAULT
CallerID	enables CallerID property in Phone control		0
CallerIDName	Type of caller ID hardware		

[System]

INI SETTING	DESCRIPTION	OPTIONS / UNITS	DEFAULT
Logs	set to 1 to enable event logging in your .exe		0
Icanfax	set to 1 to enable VBFax		0
UseInflection	set to 1 to enable use of inflection when improve quality of concatenated numbers (requires new VBVOICE.VAP)		0

[PBX]

INI SETTING	DESCRIPTION	OPTIONS / UNITS	DEFAULT
DetectDialTone	duration of nonsilence to detect as dialtone during play operations. 0 to disable.	seconds	10
MaxDialToneWait	secs to wait for dialtone before abandoning	ms	
MinDialTone	minimum period of dialtone required to start dialling	ms	
Putonhold			!,
ReconnectFromHold			!,
Xfer			!,
Connect			!,
ReconnectFromBusyNoans			!,
CallerID	Enables caller id option		False
DetectAnswerTime	Time used to detect answer machine, person in dial	ms	6000
AnswerDeglitch	silence in answer before stopping answer size detect	ms	600

[Languages]

INI SETTING	DESCRIPTION	OPTIONS / UNITS	DEFAULT
Language<n>	Name of language		

See multiple languages page I-1 for more details

[Directories]

INI SETTING	DESCRIPTION	OPTIONS / UNITS	DEFAULT
Voice	default directory for voice files		C:\VBV\
Announce	default directory for ANNOUNCE.EXE		

incoming call detected. Call abandoned

An incoming call was detected after a call had been dialled - i.e. after going offhook, dial tone was not detected. This normally happens when the system goes off hook to start dialling, just as an incoming call arrives, but before the voice card has detected ringing.

delay start

The current call has terminated on the line, and the Phone control has been set to start a call after a delay."
INFOMSG

dialling on hold digits

A call has entered a Delay control, which is now dialling the digits required to put the calle on hold during the delay.
INFOMSG

delay completed

The delay time in the Delay control has expired.
INFOMSG

reconnecting from hold

INFOMSG

waiting for digits, max n mask 0x1234

This is an information message only for those of you who know the voice card driver functions.
INFOMSG

inc count (value)

A Count control has incremented its internal counter.
INFOMSG

number to dial

A call has entered a Dial control, which has set this number to dial. Once the greeting has completed, the number will be dialled.
INFOMSG

starting dial

The greeting in a Dial control has now compteted, and is starting the dial operation, with no call progress analysis.
INFOMSG

starting call

The greeting in a Dial control has now compteted, and is starting the dial operation, with call progress analysis.
INFOMSG

number changed to (dial number)

A call entered a Dial control, and VB code changed the number to dial using the NumToDial property.
INFOMSG

transfer to (control name) via VB

VB code has set the GotoControl property in the Enter event, and the call has been transferred to a new control.
CALLFLOW

transfer to (control name) via node (number)(name)

The call has been transferred to a new control via the output node given.
CALLFLOW

value on exit is (string)

All controls that have default properties (for the purposes of Property Substitution) log this message when a call leaves the control. The value shown is the string that will be substituted for the string %ControlName% in other controls.

INFOMSG

ini (file, section, field, value)

The IniSw logs this message, showing the INI file accessed and used for branching
INFOMSG

record file name (filename)

A call has entered a Record control, which has set this filename to record into. Once the greeting has completed, the file will be created, and the record operation started.
INFOMSG

empty record file

A Record control started a record, but did not receive enough voice data for a valid voice file. You can set the minimum duration of voice required to validate a recording in the Record setup dialog. This condition will invoke the silence handler in the Record control.
INVALIDMSG

record ok

The Record control made a valid recording, and will now play the options greeting (if there is one).
INFOMSG

VB set bad block (control name)

VB code attempted to transfer a call to a new control via the GotoControl property, but the specified control does not exist.
ERRORMSG

VB set new block (control name)

VB code has set a new control to transfer to using the GotoControl property. Once VB code leaves the event, the transfer will take place.
INFOMSG

VB set bad node (node number)

VB code attempted to transfer a call to a new control via the ControlNode property, but the specified control name does not exist.
ERRORMSG

VB set new node (node number)

VB code attempted to transfer a call to a new control via the GotoNode property, but the specified node does not exist or is not connected.
INFOMSG

VB set termdigit (mask)

VB code set a new terminating digit via the TermDigit property.
INFOMSG

VB set max time (secs)

VB code set a new maximum time via the MaxTime property.
INFOMSG

VB set max sil (secs)

VB code set a new maximum silence time via the MaxSilence property.
INFOMSG

VB set new file (secs)

VB code in the Enter event for the Record control set a new filename for the file to be recorded, overriding the file selected by the Record control.
INFOMSG

VB set max keys (secs)

VB code set a new maximum number of keys for the operation via the MaxDigits property.
INFOMSG

VB set CPL file (secs)

VB code set a new CPL file using the CPLFile property in the Record control.
INFOMSG

silence, error count (secs)

An information message indicating that the control received silence while recording or waiting for digits. The error count is incremented, and if the error count is exceeded, error handling is invoked.
INVALIDMSG

invalid count (count)

An information message indicating that the control received invalid digits while waiting for digits. The error count is incremented, and if the error count is exceeded, error handling is invoked.
INVALIDMSG

no connection on node (node number, node name)

An attempt was made to transfer to another control, but the node selected for transfer is not connected, or is connected but VBVoice cannot find the destination control. Since the system is checked for complete connection integrity before system start up, this message should not normally occur except in test mode. In systems with more than one VBVoice form, you must ensure that all forms are displayed, or loaded and minimized, before starting a test. VBVoice will not be able to find controls on forms that have not been opened.

When testing in design mode, you must load the forms by selecting each form in turn and clicking View Form in the project window. The forms may be minimized if you wish.

When testing in run mode, you must tell Visual Basic to load the forms upon startup. To do this, go to the startup form (as defined in Options menu, under Project) and use a Load statement, or the WindowState property to load each form in the Form.Load procedure for that form. See the CALLANS form in the VMDEMO example for an example. Normally you will want to display the forms while testing, but have them hidden when creating your .exe. Use a command line option to hide the forms in run environment, but not in the VB environment. You can set command line options in Visual Basic environment using the Options... Project menu. To hide a form without unloading it, use the forms Visible property, or set its position off the screen.

ERRORMSG

incomplete control name (string)

While processing a field, VBVoice encountered an incomplete tag - ie a % sign with no terminating % sign.
ERRORMSG

bad_number (dial string)

A Dial control attempted to dial a number that does not contain valid characters.
ERRORMSG

bad phrase, type --- name ---

VBVoice was unable to process the phrase as defined.
ERRORMSG

vap phrase not found, file (filename), phrase (phrasename)

VBVoice was unable to process the phrase as defined. Normally due to the phrase as named does not exist in the VAP phrase specified.
ERRORMSG

play already active

This message occurs when the system attempts to play a greeting when there is already a greeting playing on the channel. This signifies that the VBVoice state machine has got out of step with reality, and should not occur.
ERRORMSG

control not found (control name)

A control name was referenced in a dialog field for substition ie using %Controlname% syntax, but the control does not exist. If in VB design mode, check that the form containing the control is showing or minimized. The controls are not loaded into the system until the form is loaded.
If VB is in run mode, ensure that the form containing the control is loaded using a Form.Load or a Form.Windowstate statement. this statement should appear in the Load event for the startup form, or in Sub Main in Global.bas, depending on what has been set in Options/Project for the startup form.
ERRORMSG

bad record filename (filename)

VBVoice was unable to open the specifed file for recording
ERRORMSG

unexpected event (event name)

VBVoice encountered an event from the voice driver for which it has no predefined response. This means either there is a bug in our voice engine (surely not), a situation on your phone line which has not been anticipated, or the voice driver itself is not working correctly. Please call Tech Support.
ERRORMSG

digits received (digit string)

The GetDigits control has terminated digit collection based on the digits received from the caller.
INFOMSG

VB set new result (string)

VB code has set a new result (default property) for the control. Applies to GetDigits, User, IniSw and DataGet controls.
INFOMSG

control not found in condition (condition name)

A GetDigits or DataGet control has a condition set using property substitution, but the specified control cannot be found. See CONTROLNOTFOUND.
ERRORMSG

data control not found

Occurs if VBVoice could not locate the data control linked to a DataFind or DataNew control at design time. Every DataFind and DataNew must reference a VB data control on the same form. This control can be selected in the setup dialog for the control.
ERRORMSG

database error (error code)

An error occurred while accessing a data control. The error code refers to a VB databaseerror - check in the VB help file for details under database error codes.
ERRORMSG

database data (fieldname, data)

Logged by a DataGet control when accessng a database field.
INFOMSG

database record found (search field, search data)

A DataFind control has found a record that matches the requirements specified.
INFOMSG

silence count exceeded (count)

A silence timeout has occurred and the control has exceeded the maximum number of timeout or invalid digit conditions allowed.
ERRORMSG

error count exceeded (count)

An invalid digit has occurred and the control has exceeded the maximum number of timeout or invalid digit conditions allowed.
ERRORMSG

branch loop count exceeded

This should not occur unless you have designed an infinite loop into your sysrtem. This is possible when using the non-voice controls - e.g. the data controls and IniSW. For instance, looping the output of a DataGet back its the main input will cause an infinite loop until stopped by this message. This message can also be generated if you string more than 30 non-voice controls together without any voice controls in between.
ERRORMSG

data not set in control (controlname)

This message occurs when an attempt is made to access a default property from a control that does not have one. Since this condition is checked at startup, this message should never happen, except when the system has been started and an attempt is made to acces a property that has not been set yet - like a forward reference., or at design time when a dialog is used to provide the default property and the Cancel button is pressed. The data provided with the message is the name of the control in question.
ERRORMSG

cannot create unique file

Generated by the Record control when it is building a new record filename. It tries 30 times to create a unique filename based on the record file specification in the dialog, and then gives up and logs this message.
ERRORMSG

database not accessible in design mode

VB data controls do not allow access to the underlying database when VB is in design mode. To allow access to your data during testing, start VB in run mode using the Run/Start menu command, and then restart your test. Note that if you have multiple forms, they must be expicitluy loaded by Visual Basic code in the Form Load event or Sub Main. See Running your application in the User's Guide.
ERRORMSG

end of database

The DataFind control came to the end of a database without finding any more records that match the criteria.
INFOMSG

bad time spec (string)

Generated by phrase play function when playing a System phraes of type Say Date, Say Time,.
ERRORMSG

vap phrase id not found (phrase number)

VBVoice is searching for a particular phrase in a VAP phrase file, but cannot find it.
ERRORMSG

bad cplfile (filename)

VBVoice cound not open the specified CPL file, or it was the wrong size.
ERRORMSG

error setting recordsource property

The Record control is updating a database with the filename recorded. An error occurred while setting the RecordSource in the data control.
ERRORMSG

restarting record

Occurs in the Record control when the caller presses the digit assigned to re-record the message.
ERRORMSG

record cancelled

Occurs in the Record control when the caller presses the digit assigned to cancel the message.
INFOMSG

playing back recording

Occurs in the Record control when the caller presses the digit assigned to play back the message.
INFOMSG

connected to xferee

The Dial control is calling a number with call progress and Usr Transfer enabled. An answer has been detected and the call continues connected to the transferee.
INFOMSG

digits dialled

The Dial control is calling a number with call progress. An answer has been detected.
INFOMSG

phrase object does not exist

The DLL function vbv_remove_userphrase was called with a bad phrase object
INFOMSG

transfer to control (control name)

Logged by the voice engine when resetting the current control for a channel back to the Phone control that owns this line.
CALLFLOW

control error

An error occurred in a control while processing a voice driver event.
ERRORMSG

search database for (data) in (fieldname)

Logged by the DataFind control when starting a database search
INFOMSG

control state (number)

Logged by all controls when changing their internal state. Refers to the state machines internal to each control which are used to manage the voice engine.
CONTROLSTATES

setting recordsource

Logged by the Record control when updating a database with a new filename.
INFOMSG

fire event failed

VBVoice received an error code when attempting to fire an event proceduer within Visual Basic
ERRORMSG

no data in control (control name)

This message occurs when an attempt is made to access a default property from a control that does not have one.
In test mode, the system will prompt you to enter some data, rather than generating this error message. If you press Cancel, this message will be generated.
At runtime, you may get this message when the system has been started and an attempt is made to acces a property that has not been set yet - like a forward reference
The data provided with the message is the string that references the control.
ERRORMSG

bookmark set

VB code set the bookmark for a DataFind control.
ERRORMSG

using cpl file (cpl filename)

A Dial control has been set to use a non-default CPL file. When the call arrives at the control, the channel will be loaded with the new Call Progress Analysis parameters before dialling and performing call analysis.
INFOMSG

transfer to (controlname) via (node number, node name)

A call has been transferred to a new control. This message logs the tranfer, and the value of the default property of the original control on exit.
CALLFLOW

voice call

Voice driver message - the voice driver CALL function has been invoked
LOGFUNCMSG

voice dial

Voice driver message - the voice driver DIAL function has been invoked
LOGFUNCMSG

voice getdtmfstring

Voice driver message - the voice driver GETDTMFSTRING function has been invoked
LOGFUNCMSG

voice play

Voice driver message - the voice driver PLAY function has been invoked
LOGFUNCMSG

voice record

Voice driver message - the voice driver RECORD function has been invoked
LOGFUNCMSG

voice offhook

Voice driver message - the voice driver SETHOOK function has been invoked
LOGFUNCMSG

voice onhook

Voice driver message - the voice driver SETHOOK function has been invoked
LOGFUNCMSG

voice getdtmf

Voice driver message - the voice driver GETDTMF function has been invoked
LOGFUNCMSG

voice clrdtmf

Voice driver message - the voice driver CLRDTMF function has been invoked
LOGFUNCMSG

voice stop

Voice driver message - the voice driver CHSTOP function has been invoked
LOGFUNCMSG

voice seize

Voice driver message - the voice driver SEIZE function has been invoked
LOGFUNCMSG

voice release

Voice driver message - the voice driver RELEASE function has been invoked
LOGFUNCMSG

voice setch

Voice driver message - the voice driver SETCH function has been invoked
LOGFUNCMSG

voice call failed (error code)

Voice driver message - the voice driver CALL function failed
ERRORMSG

voice dial failed (error code)

Voice driver message - the voice driver DIAL function failed
ERRORMSG

voice getdtmfstring failed (error code)

Voice driver message - the voice driver GETDTMFSTRING function failed
ERRORMSG

voice play failed (error code)

Voice driver message - the voice driver PLAY function failed
ERRORMSG

voice record failed (error code)

Voice driver message - the voice driver RECORD function failed
ERRORMSG

voice offhook failed (error code)

Voice driver message - the voice driver SETHOOK function failed
ERRORMSG

voice onhook failed (error code)

Voice driver message - the voice driver SETHOOK function failed
ERRORMSG

voice stop failed (error code)

Voice driver message - the voice driver CHSTOP function failed
ERRORMSG

voice seize failed (error code)

Voice driver message - the voice driver SEIZE function failed
ERRORMSG

voice release failed (error code)

voice driver message - the voice driver RELEASE function failed
ERRORMSG

voice setch failed (error code)

Voice driver message - the voice driver SETCH function failed
ERRORMSG

startup check failed in control (controlname)

A control or controls failed the startup checks. Refer to the error messages listed for more information. The system will not start until all control have been checked for setup errors, and the whole system checked for integrity.
ERRORMSG

starting test in control (controlname)

CALLFLOW

no connection on node (nodename): playing invalid entry greeting

A control has exceeded it's error count. The error output is not connected so VBVoice is playing the default error greeting prior to handing up.
CALLFLOW

inter call delay

the current call has completed and the Phone control that owns this line has been set to start another call after a delay. It has just started the delay timer.
INFOMSG

channel idle

A call has been completed and the Phone control that owns this line has gone back to idle mode.
INFOMSG

waiting for ring

A call has been completed and the Phone control that owns this line has gone back to waiting for incoming ring.
INFOMSG

playing onhold music

The Delay control has started the Delay and is playing on-hold music to the caller.
INFOMSG

VB stopped delay

VB code terminated the delay wait period in a Delay control by setting the DelayTime property
INFOMSG

loop current drop: call terminated

A loop current drop was reported by the voice card driver. This is usually due to the caller hanging up. If it occurs just after starting a call or answering a call, you may need to set the delay between going off-hook and checking for loop current. This can be changed using the INI setting OffHook_Delay - check appendix in User's Guide for more details.
CALLFLOW

bad date specification (date string)

Generated by phrase processor in System phrase - Say Date
ERRORMSG

sys start

The voice engine has been started by the Start System menu command or by the DLL function vbv_start_system
CALLS

sys stop

The voice engine has been started by the Stop System menu command or by the DLL function vbv_stop_system
CALLS

call start

A call has been started, initiated eithe by incoming ring, startcall after delay, or by VB code starting a call.
CALLS

call stop

"CALL STOP"
A call has been terminated and the line is back on-hook
CALLS

waiting for dialtone

The Dial control is waiting for dialtone before continuing to dial.
INFOMSG

updating db action

Used in PlayMsgs control
INFOMSG

get next msg

Used in PlayMsgs control to log message handling
INFOMSG

update num msgs

Used in PlayMsgs control to log message handling
INFOMSG

cannot create backup file

Used in PlayMsgs control to log message handling
ERRORMSG

mailbox file open error (maibox file)

PlayMsgs control could not open the specified mailbox file.
ERRORMSG

mailbox file write error

PlayMsgs control opened the mailbox file for read but could not write to it.
ERRORMSG

mailbox file invalid entry

PlayMsgs control could not make sense of the data in the specified mailbox file. The mailbox file has become corrupted.
ERRORMSG

voice get car failed

Voice driver error.
ERRORMSG

invalid user phrase

VB code tried to set a new phrase in a greeting but passed in an invalid phase object. Either the parameter passed in was not created using one of the vbv_create_phrase functions, or it has already been deleted.
ERRORMSG

cannot start test in VB break mode

You can start a test when Visual Basic is in design mode (with some limitations, see User's Guide, chapter on Test Mode), and also in Visual Basic run mode. You cannot start a test when in break mode - either stop Visual Basic and return to design mode, or start Visual Basic and return to run mode
ERRORMSG

vap phrase not found

The phrase name could not be found in the VAP phrase. Check that the script specified matches exactly the contents of the VAP file. Scrip matching is case sensitive.

vap phrase size was zero

Phrases selected for playing must have voice data.

file open error

The specified file could not be opened. Check the path (if supplied) is correct. If no path is supplied, the file must exist in the VBV default directory.

file size error

Vox files must be larger than 1000 bytes (250 ms) or VBVoice will not play them

no phrase data

The phrase requires parameters which are not set.

call terminated

A call has been terminated by the control.

warning: cannot generate VB event if not in run mode

The operation cannot be performed because Visual Basic is in design mode. Controls that rely on Visual Basic, such as Condition events in GetDigits, the User control or database controls do not work in design mode. Note that other controls can be tested in design mode, but that any attached code will not run until Visual Basic is put into run mode.

phone control is in idle mode, ring event ignored

An incoming ring occurred on a line, but the Phonecontrol that owns this line was in Idle mode, and ignored the Ring event.
INFOMSG

INVALIDDESTNODE

VBVoice cannot evaluate the destination connected to this output node. Please call Tech Support.

control not found

An invalid control name was specifed as a parameter in a dialog. Either the control name is incorrect, the control does not exist, or the control is on a form that is not presently loaded. If the control exists, ensure that it is on a form that is loaded (if VB is running), or on a form that is visible (if in design mode).

Control names are case sensitive, and must contain the index of the control if part of an array: i.e. ControlName(7)

file not found

The specified file could not be opened. Check the path (if supplied) is correct. If no path is supplied, the file must exist in the VBV default directory.

file error

The indicated file exists, but an error occurred while trying to access the file.

phrase error in phrase

A phrase has failed the internal checks. Check the parameters supplied with the phrase to ensure they are valid, that the phrase indicated is contained in the file, and that the file exists in the path specified.

BADBLOCKID

The destination control for an output node could not be found. If this is an OutConn control, then the control must be connected to an InConn control.

If this message occurs during system check at design time, ensure that both source and destination forms are displayed.

If the message occurs during system start, ensure that all VBVoice forms are loaded by Visual Basic before calling System Start. This can be done using Show statements in the Form.Load procedure for the main form, or Sub Main as set in Project | Options,

node not connected

As a Warning:

An output node of this control is not connected. Although it is not essential that this node be connected, ensure that this is what you want to do. Invalid digits, silence timeout and 'not found' nodes will invoke default error handlers if a call is transferred to one of these unconnected nodes. In the Phone control, the Start Call and Ring nodes may be left unconnected if the corresponding 'Wait for Ring' and 'Start Call' are not set. note, however that attempting to set these nodes at runtime will cause an error if they are not connected.

As an Error:

The system requires that this node be connected to another control.

no time period name

CHECKFAIL
"node %i, no time period name", get_node_name(node));
Each time period in a list of time periods must be named. Use the TimeSw setup dialog to add a name to each time period in the list.

no field specified

In the DataFind control, the Match All check box is not checked, but a database field to do a data search in has not been specified. Use the setup dialog to choose a field to search.
In the DataGet control, you must specify the database field from which to access the data
CHECKFAIL

no datasearch specified

In the DataFind control, the Match All check box is not checked, and a field to do a data search in has been specified, but there is no data to compare.
CHECKFAIL
"no datasearch specified"

no database table specified

In the PlayMsgs control, you must select which table of messages to access. This is normally based on a mailbox number entered by the caller and referred to using Property Substitution.

In the DataFind and DataNew controls, a data control has not been selected in which to search for data or in which to add a new record
CHECKFAIL

Feedback warning

An output node has been connected back to the input on the same control. This can cause an infinite loop to occur, or a channel to hang. Please ensure this is what you really want to do.

CHECKWARN

no valid termination specified

The GetDigits control cannot calculate valid digit termination conditions from the digit masks supplied. Add a silence timeout, or terminating digit in the GetDigits setup dialog.

CHECKWARN

digit collection will terminate on silence

The GetDigits control cannot calculate valid digit termination conditions from the digit masks supplied, and will be forced to use silence timeout only. If possible, add a terminating digit in the Setup dialog to improve response time.

max keys too small for node

The maximum number of keys specified is less than that required for the digitmask specified. For instance, if a digit mask contains nnn (3 numeric digits), and Maximum digits is 2, this error will occur. Change the digit mask or maximum digits as appropriate.
CHECKFAIL

max keys too small for node %s", condition_names[n]->gets());

cannot use wild chars in range

An invalid digit mask has been entered. The hyphen operator must be used only in conjuction with numeric digits i.e. 100-134.

You also cannot use control names in range. VBVoice does not support control names embedded into a range. Use a VB condition and add some code to do your digit matching
CHECKFAIL

range min and max must have the same number of digits

i.e 11 -23 is ok
234 - 567 is ok
23 - 567 is not ok: use 2 separate masks 23-99 and 100 - 567
CHECKFAIL

phrase error in phrase msg_recorded

the record.vap phrase does not contain the phrase your message has been recorded
CHECKFAIL,

no record filename set

The record control must have a filename set in the required format. This filename can either be a fixed file name or can contain a *, which will create a unique filename at runtime, and Control names, which will be replaced by the value of the control at run time
CHECKFAIL

no filename specified

The IniSwi control must have a fieldname specified. A .INI settting consists of 3 items, the .INI filename, the section name (the part enclosed by []), and the field name:
i.e.
filename VBVOICE.INI
sectionname [VoiceCard]
fieldname Dialogic
CHECKFAIL

no condition specified on node

A dataget or getdigits control has a condition with a name but an empty digitmask.
CHECKFAIL

no DataFind block found

A DataGet or DataChange control must be connected to a DataFind or DataNew control, either directly or via another control. VBVoice will use the first DataFind or DataNew control that it finds, so there should be only one DataNew or DataFind possible. If there are multiple paths leading to the DataGet or DataChange, VBVoice may pick the wrong one.
CHECKFAIL

data control not found

A DataFind or DataNew control must be connected to a Visual Basic control using the setup dialog.
CHECKFAIL

2 controls of name

It is possible in Visual Basic to have 2 controls of the same name on different forms. VBVoice likes to have unique names for all it's controls. Rename the control indicated to a new name. If you have other controls which reference this control, ensure these are also updated.

CHECKFAIL

Line limit exceeded

The application is not running in the Visual Basic environment, and the line number set in the Phone control exceeds the number of lines that this system has been authorized to build. For instance, a 4 line license allows the use of channesl 1-4 only. If you wish to build bigger systems, contact PRONEXUS on 613 839 0033 for an upgrade or contact your authorized distributor.

no fielddata specified

In the DataChange control, a field name has been selected for update, but no new data is attached.

no greeting

A greeting has not been specified. Although the program will work without this greeting, you should make sure this is what you want to do.

No Silence timeout greeting:

The control does not have a silence timeout greeting. this greeting is played when the control times out while waiting for user input (speech or digits). In most cases the control will play the silence timeout greeting, if configured, and then play the entry prompt again. refer to help on the particluar control for more information.

No invalid digits greeting:

The control does not have a invalid digits greeting. This greeting is played when the control receives incorrect digits from the caller. In most cases the control will play the invalid digits greeting, if configured, and then play the entry prompt again. Refer to help on the particluar control for more information.

No entry greeting:

The control does not have an entry greeting. This greeting is played when a call enters the control , and also after a silence or invalid digits greeting (see above). Normally a control will need a greeting to tell the caller what to do.

No options greeting:

The Record control does not have an options greeting. This greeting is played after a recording has been made. If you have not set any options digits, then you may not want to play an options greeting.

silence timeout = 0 but max keys > 1

In the GetDigits control, you have set the silence timeout to 0. This means that the caller is not given any time after the greeting to enter digits. However the greeting is set to terminate on the first digit, so the maximum possible number of digits that can be entered is 1. There is at least one digit mask that requires more than one digit or the max keys field is set to more than 1. Therefore there is a conflict: either increase the silence timeout, or remove the requirement for more than 1 digit as a termination.

Time Set Dialog

This dialog appears when in test mode and a TimeSw dialog is encountered. It allows you to enter the time to be used when evaluating the branch condition. Using this dialog you can test all possible branches regardless of the actual time. The default time is the actual time.

Get Value Dialog

This dialog appears when in test mode, and a control attempts to get a value from another control which has not been entered. This will happen if you have started a test from a point which has bypassed some earlier controls. Since the current control requires a value from the control specified, you must enter the value yourself.

This can be useful if you want to test different results from a database without having to search for the data you want to test with. If you select cancel, a runtime error occurs and the test will terminate.

PBX Settings Dialog

Use this dialog to enter values used by the Dial and Delay controls when accessing features in the PBX.

Directory dialog

This dialog allows you to specify the default location of the VAP phrase files and other files needed by VBVoice. It also allows you to specify the location of the Announce program. You should not normally have to change these settings.

Preopen Dialog

This dialog creates a list of files that should be opened by VBVoice before starting. Normally VBVoice will open a file when needed, and then close it again. This can lead to inefficient operation since the same files may be opened and closed over and over.

To use this dialog, you should be in design mode, and have all your forms visible. The files-used list box will show all the voice files used by your system. All files that are used frequently by your system should be moved to the pre-open list.

Error Handler Dialog

Use this dialog to set the phrases that will play when the default error handling or timeout handling in invoked.