

2005/05/06

音声認識 ActiveX コントロール

Prezam VoiceTool

for IBM ViaVoiceTM 3

Version 3.3.1048

目 次

1. コントロールのインストールとアンインストール	7
1. コントロールのインストールとアンインストール	7
2. コントロールの使用方法	8
3. リファレンス	9
プロパティ一覧	9
AppName	9
MicStatus	9
AudioSave	9
Rejection	10
Threshold	10
VV_Version	10
AudioHost	10
メソッド一覧	12
Initialize	12
InitializeEx	12
Terminate	12
Terminate Immediately	13
VocabLoad	14
VocabLoadEx	14
VocabUnload	15
VocabEnable	15
VocabDisable	15
VocabAddWord	17
VocabRemoveWord	17
MicOn	18
FocusOn	18
StartReco	19
ContReco	19
EndReco	19
ParsePhrase	20

ParsePhraseEx	20
ParseAnnotation.....	20
ParseAnnotationString.....	20
ParseAlternatives	20
PhraseVocabName	21
XtraGetPhraseNum.....	21
XtraParsePhrase	21
XtraPhraseAnnotation.....	21
XtraPhraseAnnotationString.....	22
XtraGetPhraseAlternativesNum	22
XtraGetPhraseAlternatives	22
SetPool.....	24
SetAudioSourceParam.....	24
GetUserList	25
XtraGetUserListNum	25
XtraGetUserList.....	25
XtraGetUserIdList	26
GetViaVoicePath.....	26
GetLastErrorDescription.....	26
SetHttpFtpMode.....	27
GetCurrentUserID	27
Delay	27
SetDictMode	28
SetUseAlternatives.....	28
PlayWords	28
DiscardAudioData	29
SaveWords.....	29
IsReject.....	29
VVRtkClients_Launch.....	30
VVRtkClients_IsClientAvailable	32
VVRtkClients_IsClientRunning	33
イベント一覧	34
onPhraseRecognized.....	34
onChangeMicState	34
onChangeFocusState	35
onChangeAudioState.....	35

拡張版での機能.....	36
AddPersonalPool	36
CopyPersonalVocabulary	36
CopyUserFile.....	37
4. 処理の流れ	38
5. 認識イベントの処理.....	39
6. フォーカスの移動処理	40
7. Visual Basicでのサンプル	41
8. グラマーファイルとプールファイルの作成について	43
9. 使用上の注意.....	45
10. リリースノート	46

Prezam VoiceTool for IBM ViaVoice 3 について

この度は、Prezam VoiceTool for IBM ViaVoice 3 をご利用いただきありがとうございます。

このActiveXコントロールは、IBM ViaVoiceによる音声認識をMicrosoft® Visual Basic、Webコンテンツ、Macromedia® Director®で使えるようにするためのものです。

グラマーまたは、単語を用いた Command&Control による音声認識や、口述筆記である Dictation による認識も行うことができます。

コントロールは、IBM ViaVoice の API である SMAPI(Speech Manager API)を使用して作られています。このため、次のような利点を持っています。

- (1) 処理が高速です。特に初期化処理が SAPI を用いた場合に比べて非常に高速です。
- (2) ViaVoice に対して細かな制御をしています。グラマーの候補一覧などを入手することも可能です。
- (3) ViaVoice の発音辞書の登録や FTP や HTTP プロトコルによるボキャブラリのロードも可能となっています。

動作環境

ViaVoice™ for Windows® の動作環境に準拠します。

価格および配布

価格	体験版は、無償で使用できます。エンドユーザー製品の一部として配布する際には、パーソナル版の場合、登録と同時に 148,000 円 (消費税別)をお支払いいただきます。ご登録いただけますと、フル機能をご利用できるシリアル番号を発行いたします。
2次配布	この製品を登録したエンドユーザー製品の一部として使用し、一緒に配布する場合、無償で配布できます。
拡張版	拡張版は、別途販売しております。拡張版については、お問い合わせください。上記価格はパーソナル版になります(拡張版機能を含まない)。拡張版では、パーソナルボキャブラリへの追加、パーソナルボキャブラリのコピー、ユーザーデータのコピーなどが行えるようになります。

株式会社プレザム

〒135-0032 東京都江東区福住 1-11-4 福住石橋ビル 5F

お問合せE-mail: sales@prezam.co.jp お問合せ電話番号: 03-5245-1948

サポートE-mail: support@prezam.co.jp

HomePage URL: <http://www.prezam.co.jp/>

※ViaVoice 自体は、この製品には含まれておりません。別途必要になります。

ViaVoice PowerPC は、米国 IBM Corporation の商標。

Macromedia Director Lingo は Macromedia, Inc. の登録商標。

Apple Macintosh Mac は米国 Apple Computer, Inc. の登録商標。

Microsoft Windows は、Microsoft Corporation の登録商標。

他の会社名、製品名およびサービス名等は、それぞれ各社の商標または登録商標。

『ライセンス情報』

株式会社プレザム

このライセンス情報は、『Prezam VoiceTool for IBM ViaVoice 3』（以下『本プログラム』といいます）のご使用に関するものです。

「本プログラム」をご使用になるお客様が「本プログラム」の使用を開始されると、この条件に同意されたものとみなさせていただきます（この「ライセンス情報」に同意いただけない場合は、お客様により、ただちに「プログラム」を再使用不可能な状態で廃棄してください）。

1. 著作権の帰属：

本ソフトウェアおよび本ソフトウェアに附属するマニュアル等の関連資料（以下『関連資料』といいます）に係わる著作権（キャラクターおよびその他グラフィクス、名称等も含む）およびその他一切の知的財産権は当社に帰属します。

2. 禁止事項：

お客様は、以下の行為を行うことは出来ません。

- (1) 本ソフトウェアに関し、修正、変更、改変、リバース・エンジニアリング、逆コンパイル、逆アセンブル等すること。
- (2) 本ソフトウェア若しくは関連資料又はこれらの複製物の全部若しくは一部に関して、第三者に対して販売等の利益行為を行うこと。

3. 免責：

当社は、お客様に対し、本ソフトウェア又は関連資料の欠陥、瑕疵等についてこれらを使用したこと又は使用出来なかったことから生じる一切の損害（お客様の情報の消失、毀損等による損害を含みます）に関し、損害賠償責任およびその他のいかなる責任も負わないものとします。

以上

1. コントロールのインストールとアンインストール

このコントロールのレジストリへの登録の仕方は、以下のとおりです(ウィンドウズのシステムディレクトリーが違うドライブ、パス名の場合は、そのように変えてください)。

Windows98、Windows Me の場合

C:¥Windows¥System¥RegSvr32 PZVoiceTool3.dll

Windows XP Home Edition の場合

C:¥Windows¥System32¥RegSvr32 PZVoiceTool3.dll

Windows NT、Windows 2000、Windows XP Pro の場合

C:¥WinNT¥System32¥RegSvr32 PZVoiceTool3.dll

このコントロールのレジストリからの消去方法は、以下のとおりです。

Windows98、Windows Me の場合

C:¥Windows¥System¥RegSvr32 /U PzVoiceTool3.dll

Windows XP Home Edition の場合

C:¥Windows¥System32¥RegSvr32 /U PzVoiceTool3.dll

Windows NT、Windows 2000、Windows XP Pro の場合

C:¥WinNT¥System32¥RegSvr32 /U PZVoiceTool3.dll

2. コントロールの使用方法

【コントロールの GUID など】

コントロールの CLASSID	CE5D39AF-7EAA-4B97-8E5F-02CFC7FC4D0E
タイプライブラリ名	PzVoiceTool3LibCtl
タイプライブラリ ID	C63F4724-F6E3-417E-B34D-778B2BC951F0

【Visual Basic 6.0 の場合】

Visual Basic のメニューの[プロジェクト]ー[コンポーネント...]で表示されるダイアログで、**Prezam VoiceTool for IBM ViaVoice 3** をチェックします。そして、[適用]ボタンで適用して[OK]ボタンで閉じます。マイクの形をしたアイコン(SpeechCtrl)をドラッグしてフォームに貼り付けます。

現在、このコントロールは、ひとつのフォームにひとつしか貼り付けることができません。

【Director の場合】

Director のメニューの[挿入]ー[コントロール]ー[ActiveX...]で表示されるダイアログで、**Prezam VoiceTool for IBM ViaVoice 3** を選択します。

すると、ActiveX コントロールのプロパティが表示されます。ここで、[OK]を選択すると、内蔵キャストに追加されます。あとは、このエクストラをスコアウィンドウにドラッグしてスプライトにすれば使用可能です。Director は、スプライトの最初で ActiveX コントロールを使用し、スプライトの終わりでコントロールをクローズします。ですから、スコアの最初と最後までスプライトを伸ばして使用してください。

3. リファレンス

プロパティ一覧

AppName

Property AppName As String

アプリケーション名を設定します。これはViaVoiceが内部的に使用するものです。
read/write可能です。

<Visual Basicでの例>

(セットの例) SpeechCtrl1.AppName = "MyApp"

(取得の例) ap_name = SpeechCtrl1.AppName

<Directorでの例>

(セットの例) Set the AppName of Sprite x to "MyApp"

(取得の例) Put the AppName of Sprite x into ap_name

MicStatus

Property MicStatus As Boolean

現在のViaVoiceのマイクの状態を表します。falseの場合は、マイクがオフになっています。trueの場合は、マイクがオンの状態になっています。readのみ可能です。

<Visual Basicでの例>

(取得の例) mic_status = SpeechCtrl1.MicStatus

<Directorでの例>

(取得の例) Put the MicStatus of Sprite x into mic_status

AudioSave

Property AudioSave As Boolean

ViaVoiceに対して、認識中の情報をセーブするかを設定します。通常はfalseにします。一部のDictationで使用する関数を呼ぶために必要となります。read/write可能です。

<Visual Basicでの例>

(セットの例) SpeechCtrl1.AudioSave = true

(取得の例) SaveAudio = SpeechCtrl1.AudioSave

<Directorでの例>

(セットの例) Set the AudioSave of Sprite x to true

(取得の例) Put the AudioSave of Sprite x into SaveAudio

Rejection

Property Rejection As Boolean

ViaVoiceのエンジンが認識した単語を拒否した場合に、その認識単語を捨てるかを設定します。これをfalseにすると、認識し易くなりますが、雑音などを拾いやすくなります。read/write可能です。デフォルトはtrueです。

<Visual Basicでの例>

(セットの例) SpeechCtrl1.Rejection = true

(取得の例) Rejection = SpeechCtrl1.Rejection

<Directorでの例>

(セットの例) Set the Rejection of Sprite x to true

(取得の例) Put the Rejection of Sprite x into Rejection

Threshold

Property Threshold As Integer

ViaVoiceに対して、はボキャブラリにない単語またはバックグラウンドのノイズの拒否する度合いを設定します。0～10の間で設定してください。余り高くすると認識が厳しくなりますし、低くしすぎるとノイズを拾いやすくなります。read/write可能です。デフォルトはコントロールパネルで設定されている値になります。

<Visual Basicでの例>

(セットの例) SpeechCtrl1.Threshold = 5

(取得の例) Threshold = SpeechCtrl1.Threshold

<Directorでの例>

(セットの例) Set the Threshold of Sprite x to 5

(取得の例) Put the Threshold of Sprite x into Threshold

VV_Version

Property VV_Version As String

ViaVoiceのバージョン番号です。readのみ可能です。

<Visual Basicでの例>

(取得の例) version = SpeechCtrl1.VV_Version

<Directorでの例>

(取得の例) Put the VV_Version of Sprite x into version

AudioHost

Property AudioHost As String

ViaVoiceに接続する際のAudioHostストリングです。通常のマイク入力ではヌル文

字列を設定します。規定値はヌル文字列(“”)です。カスタムDLLによりWavファイルからの入力などを行う際には次のような構文で文字列を定義します。

type;dllname;key;init_string

例えばViaVoice Version 8以降に付属しているカスタムオーディオDLLを指定するときには次のようにします。“wav;audfile.dll;wav;gabage”

InitializeメソッドをCallする前にセットしてください。カスタムDLLの使用を止めて、デフォルトのマイク入力に戻したいときは、ヌル文字列(“”)を設定します。

<Visual Basicでの例>

(セットの例) SpeechCtrl1. **AudioHost** = “wav;audfile.dll;wav;gabage”

(取得の例) AudioHost = SpeechCtrl1.**AudioHost**

<Directorでの例>

(セットの例) **Set the** AudioHost **of** **Sprite** x **to** “wav;audfile.dll;wav;gabage”

(取得の例) **Put the** AudioHost **of** **Sprite** x **into** AudioHost

メソッド一覧

メソッドの戻り値は、SMAPIからのエラーコードになります。

■ 初期化/終了処理

Initialize

long Initialize()

ViaVoice音声認識エンジンに接続します。最初に一度だけ必ず呼び出します。

Directorの場合、ムービーの最初で必ず一度だけ呼ぶ必要があります。

<Visual Basicでの例>

(呼びだし例) `SpeechCtrl1.Initialize`

<Directorでの例>

(呼びだし例) `Initialize(Sprite x)`

戻り値として-30000が返った場合には、ViaVoiceがインストールされていない可能性があります。

InitializeEx

long InitializeEx(BSTR bstrUserID, BSTR bstrEnrollID)

ViaVoice音声認識エンジンに接続します。最初に一度だけ必ず呼び出します。

InitializeExは、ユーザーIDとエンロールIDを指定することができます。エンロールIDがNULLの場合には、デフォルトのエンロールIDが使われます。

Directorの場合、ムービーの最初で必ず一度だけ呼ぶ必要があります。

<Visual Basicでの例>

(呼びだし例) `SpeechCtrl1.InitializeEx("user_id", "enroll_id")`

<Directorでの例>

(呼びだし例) `InitializeEx(Sprite x, "userid", "enrolled")`

戻り値として-30000が返った場合には、ViaVoiceがインストールされていない可能性があります。

Terminate

long Terminate()

ViaVoice音声認識エンジンとの接続を解除します。

Directorの場合、ムービーの終了時に必ず一度だけ呼ぶ必要があります。

<Visual Basicでの例>

(呼びだし例) `SpeechCtrl1.Terminate`

<Directorでの例>

(呼びだし例) `Terminate(Sprite x)`

Terminate Immediately

long Terminate Immediately ()

ViaVoice音声認識エンジンとの接続を即時に解除します。このファンクションは、プールファイルのコピー時など、ViaVoiceのエンジンが終了している必要があるときに使用します。ただし、他にViaVoiceを使用しているアプリケーションがあると、ViaVoiceエンジンは終了しません。

Directorの場合、ムービーの終了時に必ず一度だけ呼ぶ必要があります。

<Visual Basicでの例>

(呼びだし例) `SpeechCtrl1.TerminateImmediately`

<Directorでの例>

(呼びだし例) `TerminateImmediately(Sprite x)`

■ グラマーボキャブラリ処理

VocabLoad

long VocabLoad(BSTR VocabName, BSTR bstrPathname)

グラマーボキャブラリをロードします。

パラメータ1は、ボキャブラリ名です。ボキャブラリ名は、この後使用する

VocabUnload、VocabEnable、VocabDisableを使用する際の識別子となります。

パラメータ2は、グラマーファイルのパス名を指定します。パス名はフルパスで指定しないと、カレントからのパスになります。また、このメソッドを複数回呼び出して、複数のボキャブラリーをロードすることも可能です。通常、Directorをスタートメニューから起動したときのカレントパスは、「c:¥windows」です。アプリケーションの起動パスを取り出す方法については、サンプルプログラムをご覧ください。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1.VocabLoad("VOCAB_NAME ", " C:¥MyApp¥Grammar.fsg ")

<Directorでの例>

(呼びだし例) VocabLoad(Sprite x, "VOCAB_NAME", "C:¥MyApp¥Grammar.fsg")

2度目以降callして戻り値として40000が返った場合には、体験版のためです。
registe_を行なってください。

ファイル名には次のような指定も可能です。

http://www.myhomepage.ne.jp/poolfile/sample.fsg

ftp://ftp.myhomepage.ne.jp/poolfile/sample.fsg

file:///C:/Documents%20and%20Settings/voice/poolfile/sample.fsg

VocabLoadEx

long VocabLoadEx(BSTR bstrVocabName, BSTR bstrPathname, long type)

ボキャブラリをロードします。ボキャブラリは、グラマーボキャブラリまたは、単語ボキャブラリです。

パラメータ1は、ボキャブラリ名です。ボキャブラリ名は、この後使用する

VocabUnload、VocabEnable、VocabDisableを使用する際の識別子となります。

パラメータ2は、グラマーファイルのパス名を指定します。パス名はフルパスで指定しないと、カレントからのパスになります。また、このメソッドを複数回呼び出して、複数のボキャブラリーをロードすることも可能です。通常、Directorをスタートメニューから起動したときのカレントパスは、「c:¥windows」です。

パラメータ3は、ロードするボキャブラリのタイプを指定します。type=1の場合、単語リストファイル、type=2の場合、グラマーボキャブラリを表します。

<Visual Basicでの例>

(呼びだし例) `SpeechCtrl1.VocabLoadEx("VOCAB_NAME", "C:¥MyApp¥word.txt", 1)`

<Directorでの例>

(呼びだし例) `VocabLoadEx(Sprite x, "VOCAB_NAME", "C:¥MyApp¥Grammar.fsg", 2)`

2度目以降callして戻り値として40000が返った場合には、体験版のためです。
`registe_`を行なってください。

ファイル名には次のような指定も可能です。

`http://www.myhomepage.ne.jp/poolfile/sample.fsg`

`ftp://ftp.myhomepage.ne.jp/poolfile/sample.fsg`

`file:///C:/Documents%20and%20Settings/voice/poolfile/sample.fsg`

VocabUnload

long VocabUnload(BSTR bstrVocabName)

グラマーボキャブラリをアンロードします。

パラメータには、VocabLoadで指定したボキャブラリ名を指定します。

<Visual Basicでの例>

(呼びだし例) `SpeechCtrl1.VocabUnLoad("VOCAB_NAME")`

<Directorでの例>

(呼びだし例) `VocabUnload(Sprite x, "VOCAB_NAME")`

VocabEnable

long VocabEnable(BSTR bstrVocabName)

ボキャブラリ名で指定したグラマーボキャブラリーをViaVoiceの認識の対象にします。

このメソッドを複数回呼び出して、同時に複数のボキャブラリーを認識の対象にすることもできます。パラメータには、VocabLoadで指定したボキャブラリ名を指定します。

<Visual Basicでの例>

(呼びだし例) `SpeechCtrl1.VocabEnable("VOCAB_NAME")`

<Directorでの例>

(呼びだし例) `VocabEnable(Sprite x, "VOCAB_NAME")`

VocabDisable

long VocabDisable(BSTR bstrVocabName)

ボキャブラリ名で指定したグラマーボキャブラリーをViaVoiceの認識の対象から外します。

パラメータには、VocabLoadで指定したボキャブラリ名を指定します。

<Visual Basicでの例>

(呼びだし例) `SpeechCtrl1.VocabDisable("VOCAB_NAME")`

<Directorでの例>

(呼びだし例) `VocabDisable(Sprite x, "VOCAB_NAME")`

VocabAddWord

long VocabAddWord(BSTR bstrVocabName, BSTR addWords[])

ボキャブラリ名で指定したボキャブラリーに単語を追加します追加できるボキャブラリは、単語リストとディクテーション用のtextボキャブラリだけです。

パラメータ1には、VocabLoadで指定したボキャブラリ名を指定します。

パラメータ2には、追加する単語の配列を指定します。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. **VocabAddWord** ("VOCAB_NAME", words)

DirectorなどActiveXに対する配列を扱えないものでは呼び出すことができません

VocabRemoveWord

long VocabRemoveWord(BSTR bstrVocabName, BSTR removeWords[])

ボキャブラリ名で指定したボキャブラリーから追加した単語を取り除きます。ボキャブラリは、単語リストとディクテーション用のtextボキャブラリだけです。

パラメータ1には、VocabLoadで指定したボキャブラリ名を指定します。

パラメータ2には、取り除く単語の配列を指定します。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. **VocabRemoveWord** ("VOCAB_NAME", words)

DirectorなどActiveXに対する配列を扱えないものでは呼び出すことができません

■ マイク/マイクフォーカス処理

MicOn

long MicOn(VARIANT_BOOL bOn)

ViaVoiceのマイクをオンまたはオフの状態にします。マイクがオンになっていないと、ViaVoiceIにはマイクから音声が入りません。
パラメータには、マイクのオン/オフを指定します。オンにする場合にはtrueを指定します。オフにする場合にはfalseを指定します。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. **MicOn** (true)

<Directorでの例>

(呼びだし例) **MicOn** (Sprite x, true)

FocusOn

long FocusOn(VARIANT_BOOL bOn)

マイクフォーカスを自分に向けるか、他の音声認識アプリに向けるかを指定します。
通常、自分のウィンドウがアクティブになっているときにフォーカスを取得して、他のアプリケーションのウィンドウがアクティブなときにはフォーカスをリリースします。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. **FocusOn** (true)

<Directorでの例>

(呼びだし例) **FocusOn** (Sprite x, true)

■認識開始/終了処理

StartReco

long StartReco()

認識を開始します。マイクとフォーカスがオンになっていないと、認識開始しても結果は返りません。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. StartReco()

<Directoreでの例>

(呼びだし例) StartReco(Sprite x)

ContReco

long ContReco()

認識を再開します。語句認識イベントが発生すると、一度認識を停止します。認識イベントで、結果の処理を行い、ボキャブラリーの変更などが終了し、認識を続けられる状態になったときに呼び出します。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. ContReco()

<Directoreでの例>

(呼びだし例) ContReco(Sprite x)

EndReco

long EndReco()

認識を終了します。これ以上認識を続ける必要の無くなった場合に呼び出します。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. EndReco()

<Directoreでの例>

(呼びだし例) EndReco(Sprite x)

■認識結果の取得処理

ParsePhrase

VARIANT ParsePhrase()

認識した単語の配列を取得します。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. [ParsePhrase\(\)](#)

DirectorなどActiveXに対する配列を扱えないものでは呼び出すことができません。

そのような場合には、Xtraが頭についたメソッドをご利用ください。

ParsePhraseEx

VARIANT ParsePhraseEx(long tag[])

認識した単語の配列とタグ情報の配列を取得します。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. [ParsePhraseEx\(tag\)](#)

DirectorなどActiveXに対する配列を扱えないものでは呼び出すことができません。

そのような場合には、Xtraが頭についたメソッドをご利用ください。

ParseAnnotation

VARIANT ParseAnnotation()

認識した単語の数値アノテーション情報配列を取得します。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. [ParseAnnotation\(\)](#)

DirectorなどActiveXに対する配列を扱えないものでは呼び出すことができません。

そのような場合には、Xtraが頭についたメソッドをご利用ください。

ParseAnnotationString

VARIANT ParseAnnotationString()

認識した単語の文字列アノテーション情報配列を取得します。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. [ParseAnnotationString\(\)](#)

DirectorなどActiveXに対する配列を扱えないものでは呼び出すことができません。

そのような場合には、Xtraが頭についたメソッドをご利用ください。

ParseAlternatives

VARIANT ParseAlternatives()

認識した単語の候補単語配列を取得します。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. Alternatives()

DirectorなどActiveXに対する配列を扱えないものでは呼び出すことができません。

そのような場合には、Xtraが頭についたメソッドをご利用ください。

PhraseVocabName

BSTR PhraseVocabName()

認識した単語の属するボキャブラリ名を取得します。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. PhraseVocabName ()

<Directoreでの例>

(呼びだし例) PhraseVocabName(Sprite x)

XtraGetPhraseNum

long XtraGetPhraseNum()

認識した語句の単語数を取得します。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. XtraGetPhraseNum ()

<Directoreでの例>

(呼びだし例) XtraGetPhraseNum(Sprite x)

XtraParsePhrase

BSTR XtraParsePhrase(long word_index)

認識した単語を取得します。パラメータには何番目の単語を得するかを指定します。インデックスは0から始まります。インデックスの最大値は、XtraGetPhraseNumメソッドで取得した値-1の値までです。たとえば、XtraGetPhraseNumで2が返ってきた場合には、インデックスに指定できるのは、0と1です。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. XtraParsePhrase (word_index)

<Directoreでの例>

(呼びだし例) XtraParsePhrase(Sprite x, word_index)

XtraPhraseAnnotation

BSTR XtraPhraseAnnotation(long word_index)

認識した単語の数値アノテーション情報を取得します。パラメータには何番目の単語の情報を取得するかを指定します。インデックスは0から始まります。インデックスの最大値は、XtraGetPhraseNumメソッドで取得した値-1の値までです。たとえば、XtraGetPhraseNumで2が返ってきた場合には、インデックスに指定できるのは、0と1です。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. [XtraPhraseAnnotation](#) (word_index)

<Directoreでの例>

(呼びだし例) [XtraPhraseAnnotation](#)([Sprite](#) x, word_index)

XtraPhraseAnnotationString

BSTR XtraPhraseAnnotationString(long word_index)

認識した単語の文字列アノテーション情報を取得します。パラメータには何番目の単語の情報を取得するかを指定します。インデックスは0から始まります。インデックスの最大値は、XtraGetPhraseNumメソッドで取得した値-1の値までです。たとえば、XtraGetPhraseNumで2が返ってきた場合には、インデックスに指定できるのは、0と1です。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. [XtraPhraseAnnotationString](#) (word_index)

<Directoreでの例>

(呼びだし例) [XtraPhraseAnnotationString](#)([Sprite](#) x, word_index)

XtraGetPhraseAlternativesNum

long XtraGetPhraseAlternativesNum()

認識した語句の候補単語の数を取得します。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. [XtraGetPhraseAlternativesNum](#) ()

<Directoreでの例>

(呼びだし例) [XtraGetPhraseAlternativesNum](#)([Sprite](#) x)

XtraGetPhraseAlternatives

BSTR XtraGetPhraseAlternatives(long word_index)

認識した語句の候補単語を取得します。パラメータには何番目の単語を取得するかを指定します。インデックスは0から始まります。インデックスの最大値は、XtraGetPhraseAlternativesNumメソッドで取得した値-1の値までです。たとえば、

XtraGetPhraseAlternativesNumで2が返ってきた場合には、インデックスに指定できるのは、0と1です。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. [XtraGetPhraseAlternatives](#) (word_index)

<Directoreでの例>

(呼びだし例) [XtraGetPhraseAlternatives](#)([Sprite](#) x, word_index)

■その他の処理

SetPool

long SetPool(BSTR bstrPathName, BSTR bstrLang, long override)

ViaVoiceのプールフォルダーに指定されたプールファイルをコピーします。グラマーなどを利用したCommand&Control認識では、発音プールが必要になります。

発音プールはViaVoiceエンジン起動時の初期化処理で読み込まれます。そのためViaVoiceエンジンを起動する前にコピーする必要があります。通常はアプリケーションのインストール時に一緒にインストールすることをお勧めします。

このSetPoolメソッドは、Initializeメソッドを呼び出す前に行ってください。Initializeを行うとエンジンが起動されてしまいます。

・bstrPathNameにはプールファイルをフルパスで指定します。コピー先のファイル名はbstrPathNameのフォルダー名を除いたファイル名になります。**他のアプリケーションと名前がぶつからないようなファイル名にするようにしてください。**

・bstrLangには、日本語の場合"Ja_JP"、米国英語の場合"En_US"を指定します。

・overrideには、次の値を指定します。

0: 上書き禁止

1: コピー先ファイルが無い、コピー元ファイルの日付が新しい場合はコピー

2: 強制的にコピー

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. SetPool("c:\poolfile", "Ja_JP", 1)

<Directorでの例>

(呼びだし例) SetPool (Sprite x, "c:\poolfile", "Ja_JP", 1)

ファイル名には次のような指定も可能です。

<http://www.myhomepage.ne.jp/poolfile/sample.ppl>

<ftp://ftp.myhomepage.ne.jp/poolfile/sample.ppl>

<file:///C:/Documents%20and%20Settings/voice/poolfile/sample.ppl>

SetAudioSourceParam

long SetAudioSourceParam (BSTR bstrParam)

カスタムオーディオDLLなどのAudioSourceに対してパラメータを指定するために使用します。wavファイルを入力にして音声認識する場合には、SetAudioHostプロパティでカスタムDLLを設定してからInitializeメソッドをCallします。実際に認識するwavファイルをこのSetAudioSourceParamメソッドで指定してから、認識の開始とマイクのオンを行うと認識されます。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. **SetAudioSourceParam** ("c:¥script.wav")

<Directorでの例>

(呼びだし例) **SetAudioSourceParam** (Sprite x, "c:¥script.wav")

GetUserList

VARIANT GetUserList(VARIANT* pUserIdList)

ViaVoiceに設定されているユーザー名とユーザーIDの一覧を取得します。ユーザーIDは引数に指定されている変数に格納されます。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. **GetUserList** (pUserList)

DirectorなどActiveXに対する配列を扱えないものでは呼び出すことができません。
そのような場合には、Xtraが頭についたメソッドをご利用ください。

XtraGetUserListNum

long XtraGetUserListNum ()

ViaVoiceに設定されているユーザーの数を取得します。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. **XtraGetUserListNum** ()

<Directorでの例>

(呼びだし例) **XtraGetUserListNum** (Sprite x)

XtraGetUserList

BSTR XtraGetUserList (long index)

ViaVoiceに設定されているユーザー名を取得します。パラメータには何番目のユーザー名を取得するか指定します。インデックスは0から始まります。インデックスの最大値は、XtraGetUserListNumメソッドで取得した値-1の値までです。たとえば、XtraGetUserListNumで2が返ってきた場合には、インデックスに指定できるのは、0と1です。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. **XtraGetUserList** (index)

<Directorでの例>

(呼びだし例) **XtraGetUserList** (Sprite x, index)

XtraGetUserIdList

BSTR XtraGetUserIdList (long index)

ViaVoiceに設定されているユーザーIDを取得します。パラメータには何番目のユーザーIDを取得するか指定します。インデックスは0から始まります。インデックスの最大値は、XtraGetUserListNumメソッドで取得した値-1の値までです。たとえば、XtraGetUserListNumで2が返ってきた場合には、インデックスに指定できるのは、0と1です。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. XtraGetUserIdList (index)

<Directorでの例>

(呼びだし例) XtraGetUserIdList (Sprite x, index)

GetViaVoicePath

BSTR GetViaVoicePath ()

ViaVoiceの入っているフォルダーのパスを返します。これはViaVoiceフォルダーの中のルートになります。C:¥Program Files¥ViaVoice配下に入っている場合にはそのままC:¥Program Files¥ViaVoiceが返ってきます。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. GetViaVoicePath ()

<Directorでの例>

(呼びだし例) GetViaVoicePath (Sprite x)

GetLastErrorDescription

long GetLastErrorDescription ()

最後にエラーになった原因の記述を返します。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. GetLastErrorDescription ()

<Directorでの例>

(呼びだし例) GetLastErrorDescription (Sprite x)

SetHttpFtpMode

long SetHttpFtpMode (long mode)

ボキャブラリのロードとプールファイルのコピーを行うときに、
ftp://ftp.myhomepage.ne.jp/poolfile/sample.fsg などと書かれたときにFTPで直接取得するか、WinInetのHTTPへの変換機能を使用するかモード設定するために使用します。規定値は、1で、FTPを直接発行するモードです。0にするとHTTPへの変換機能を使用するようになります。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. SetHttpFtpMode (1)

<Directorでの例>

(呼びだし例) SetHttpFtpMode (Sprite x, 1)

GetCurrentUserID

BSTR GetCurrentUserID ()

現在設定されているユーザーIDを返します。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. GetCurrentUserID ()

<Directorでの例>

(呼びだし例) GetCurrentUserID (Sprite x)

Delay

long Delay (long MiliSec)

指定された時間(ミリ秒)経つと戻ります(ディレイ)。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. Delay (100)

<Directorでの例>

(呼びだし例) Delay (Sprite x, 100)

SetDictMode

long SetDictMode(long mode)

ディクテーションの際に認識結果をひらがなで返すか、通常通りの漢字まじりの文字列で返すかを設定します。デフォルトでは、漢字まじりの文字列になります。

mode = 0のとき漢字まじり、mode=1のときひらがなになります。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. SetDictMode (1)

<Directorでの例>

(呼びだし例) SetDictMode (Sprite x, 1)

SetUseAlternatives

long SetUseAlternatives (VARIANT_BOOL bUse)

認識結果の候補を後で取得して使用するかを設定します。デフォルトでは、認識候補を使用するになっています。候補列を使用しないことにより、認識結果から候補列を取得する処理が無くなるため、認識速度が少し速くなります。

候補列を使用する場合Trueにします。候補は必要無くて、少しでも認識速度を早くしたい場合には、Falseにします。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. SetUseAlternatives (False)

<Directorでの例>

(呼びだし例) SetUseAlternatives (Sprite x, False)

PlayWords

long PlayWords (VARIANT* tags, long option)

タグ配列で指定された音声を再生します。音声を再生するためには、マイクがオフになっている必要があります。

optionには、0 あるいは、単語間の無音部分も再生したい場合には4を設定してください。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. PlayWords (tags, option)

DiscardAudioData

long DiscardAudioData ()

アプリケーション用にセーブされている認識した音声データを破棄します。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. [DiscardAudioData](#) ()

SaveWords

long SaveWords (VARIANT* tags, BSTR bstrSaveWavFilename)

タグ配列で指定された音声を指定されたファイルにWAVEファイルとして保存します。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. [SaveWords](#)(tags, "C:¥save.wav")

IsReject

VARIANT_BOOL IsReject (VARIANT* tags, BSTR bstrSaveWavFilename)

直前の認識単語がエンジンの中でリジェクトされていたかを返します。この関数は
文法認識とディクテーション認識の時のみ正しい値を返します。また、
Rejectionプロパティがtrueの場合には、意味がありません。

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. [IsReject](#)()

■ ViaVoiceランタイムクライアント起動処理

ViaVoiceランタイムクライアント起動処理は、ViaVoice Version 8から追加された機能です。ですので、この機能呼び出して処理されるのは、ViaVoice Version 8以降になります。これ以前のバージョンで実行した場合は、コントロール側で何もせずに戻します。この機能が入っているViaVoiceのバージョンかどうかは、VV_Versionプロパティにより取得したViaVoiceのバージョンにより判断するようにしてください。

VVRtkClients_Launch

long VVRtkClients_Launch(long IClientID, BSTR bstrOptions)

ランタイムクライアントを起動します。IClientIDに起動するクライアントのIDを指定します。bstrOptionsには、それぞれのクライアントに渡すオプションを指定します。

IClientID

- 0: ユーザーウィザード
- 1: オーディオセットアップ
- 2: エンロール
- 3: ボキャブラリエクスパンダー
- 4: ViaVoiceのオプション
- 5: ディクテーションマクロエディタ
- 6: ボキャブラリーマネージャー
- 7: ユーザー移行ツール
- 8: ボキャブラリートピックインストールツール

bstrOptions

全てのランタイムクライアントで、オプションに言語を指定する必要があります。言語は次のように指定します。

-Language=<Lang> **※必須**

<Lang>は、日本語の場合Ja_JP、米国英語の場合En_USなど言語を指定します。

◆ユーザーウィザード(0)

-User=<value> **※必須**

N – デフォルトユーザーが入った状態で入力も可能

D – ユーザーは空で、入力する

S – デフォルトユーザーが入った状態で既存のユーザーの選択のみ

-LaunchAudioSetup

ウィザードでオーディオセットアップが行なえるようになります。

-LaunchVocabExpander

ウィザードでボキャブラリエクスパンダーが表示されます(実際には無い)。

-LaunchQuickTraining

ウィザードでクイックエンロールが行なえるようになります。

-LaunchUnsupervisedEnrollment

ウィザードでスーパーバイズドエンロールが行なえるようになります。

-DisplayLastPanel

最後に表示していたパネルを表示します。

◆オーディオセットアップ(1)

-RestoreMicLevel

◆エンロール(2)

-User=S

-LaunchEnrollment

◆ボキャブラリエクスパンダー(3)

オプションはありません

◆ViaVoiceのオプション(4)

-StartPage=<value> where <value> can be:

4 - User page

16 - Voice page

1024 - Formatting 1 page

2048 - Formatting 2 page

◆ディクテーションマクロエディタ(5)

オプションはありません

◆ボキャブラリーマネージャー(6)

オプションはありません

◆ユーザー移行ツール(7)

オプションはありません

◆ボキャブラリートピックインストールツール(8)

オプションはありません

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1.VVRtkClients_Launch(1, "-Language=Ja_JP -RestoreMicLevel")

<Directoreでの例>

(呼びだし例) VVRtkClients_Launch (Sprite x, 1, "-Language=Ja_JP -RestoreMicLevel")

リターンコード

- 0: VVRTKRC_OK
- 1: VVRTKRC_CLIENT_IS_INVALID
- 2: VVRTKRC_CLIENT_OPTION_IS_INVALID
- 3: VVRTKRC_CLIENT_OPTION_IS_MISSING
- 4: VVRTKRC_CLIENT_IS_NOT_FOUND
- 5: VVRTKRC_CLIENT_IS_NOT_RUNNING
- 6: VVRTKRC_CLIENT_IS_ALREADY_RUNNING
- 7: VVRTKRC_CLIENT_INSUFFICIENT_MEMORY
- 8: VVRTKRC_CLIENT_LAUNCH_FAILED
- 9: VVRTKRC_CLIENT_RUNTIME_PATH_UNAVAILABLE

VVRtkClients_IsClientAvailable

long VVRtkClients_IsClientAvailable(long IClientID)

指定したランタイムクライアントがインストールされているか調べます。IClientIDにインストールされているか調べたいクライアントのIDを指定します。

IClientID

- 0: ユーザーウィザード
- 1: オーディオセットアップ
- 2: エンロール
- 3: ボキャブラリエクスパンダー
- 4: ViaVoiceのオプション
- 5: ディクテーションマクロエディタ
- 6: ボキャブラリーマネージャー
- 7: ユーザー移行ツール
- 8: ボキャブラリートピックインストールツール

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1.VVRtkClients_IsClientAvailable (1)

<Directoreでの例>

(呼びだし例) VVRtkClients_IsClientAvailable (Sprite x, 1)

VVRtkClients_IsClientRunning

long VVRtkClients_IsClientRunning (long IClientID)

指定したランタイムクライアントが起動されているか調べます。IClientIDに起動されているか調べたいクライアントのIDを指定します。

IClientID

- 0: ユーザーウィザード
- 1: オーディオセットアップ
- 2: エンロール
- 3: ボキャブラリエクスパンダー
- 4: ViaVoiceのオプション
- 5: ディクテーションマクロエディタ
- 6: ボキャブラリーマネージャー
- 7: ユーザー移行ツール
- 8: ボキャブラリートピックインストールツール

<Visual Basicでの例>

(呼びだし例) SpeechCtrl1. VVRtkClients_IsClientRunning (1)

<Directoreでの例>

(呼びだし例) VVRtkClients_IsClientRunning (Sprite x, 1)

イベント一覧

■ 語句認識イベント

onPhraseRecognized

onPhraseRecognized()

ViaVoiceでグラマーに入っている語句を認識すると発生するイベントハンドラーです。ハンドラーには、認識した語句がひとつになったテキストが渡されます。渡された語句のみでよければ、このテキストを使用します。語句を単語単位にバラバラに取得したい場合には、このハンドラー内で認識結果の取得処理用のメソッドを呼び出して取得することになります。

(ハンドラー例)

<Visual Basicでの例>

```
Private Sub SpeechCtrl1_OnPhraseRecognized(ByVal Phrase As String)
```

```
End Sub
```

<Directoreでの例>

```
on onPhraseRecognized text
```

```
end
```

■ マイクの状態変化イベント

onChangeMicState

onChangeMicState()

マイクがオンからオフあるいはオフからオンになると発生するイベントです。通常、マイクのインジケータに使用します。ハンドラーには、マイクの状態がついてきます。trueの時には、マイクがオンの状態に変化し、falseのときには、マイクがオフの状態に変化したことを表します。

(ハンドラー例)

<Visual Basicでの例>

```
Private Sub SpeechCtrl1_OnChangeMicState (ByVal bOn As Boolean)
```

```
End Sub
```

<Directoreでの例>

```
on onChangeMicState state
```

```
end
```

■フォーカスの状態変化イベント

onChangeFocusState

onChangeFocusState()

フォーカスがインアクティブからアクティブにあるいはアクティブからインアクティブになると発生するイベントです。ハンドラーには、フォーカスの状態がついてきます。trueの時には、フォーカスがアクティブの状態に変化し、falseのときには、フォーカスがオフの状態に変化したことを表します。

(ハンドラー例)

<Visual Basicでの例>

```
Private Sub SpeechCtrl1_OnChangeFocusState (ByVal bOn As Boolean)
```

```
End Sub
```

<Directorでの例>

```
on onChangefocusState state
```

```
end
```

■オーディオの状態変化イベント

onChangeAudioState

onChangeAudioState()

マイクからの入力レベルが変化すると発生するイベントです。ハンドラーには、オーディオレベルがついてきます。オーディオレベルは0から10までの値になります。

(ハンドラー例)

<Visual Basicでの例>

```
Private Sub SpeechCtrl1_OnChangeAudioState (ByVal lLevel As Long)
```

```
End Sub
```

<Directorでの例>

```
on onChangeAudioState level
```

```
end
```

拡張版での機能

(拡張版を購入しないと使えません)

AddPersonalPool

long AddPersonalPool(BSTR bstrWord, BSTR bstrSoundlike, BSTR BaseForm, long flag)

ViaVoiceのパーソナルボキャブラリに指定された単語を登録します。このメソッドで登録すると、以降ディクテーションでその単語も使えるようになります。発音プールが作成されますので、ナビゲーションでもプールファイルに無くてもボキャブラリのロードができるようになります。Wordには「単語」をSoundlikeには単語の「よみ」を入れます。日本語版のViaVoiceの場合には、BaseFormに何も入ってなくても「よみ」から自動生成します。ただし、余りにも不自然な「よみ」の場合にはエラーになります。英語版のViaVoiceの場合には、BaseFormは必須です。なおflagはリザーブです。0を設定してください。

<Visual Basicでの例>

(呼びだし例) `SpeechCtrl1.AddPersonalPool("KONISHIKI", "ここにしき", "", 0)`

<Directorでの例>

(呼びだし例) `Initialize(Sprite x, "KONISHIKI", "ここにしき", "", 0)`

CopyPersonalVocabulary

long CopyPersonalVocabulary(BSTR bstrUserID, BSTR bstrSrcDirectory)

ViaVoiceのパーソナルボキャブラリを指定されたディレクトリから指定されたユーザーIDのディレクトリにコピーします。指定されたユーザーIDのディレクトリが無い場合は自動生成します。コピー元としてはローカルディレクトリあるいは、FTPサーバーのディレクトリが指定できます。

注意が必要なのは、元のディレクトリにはあるファイルをすべてコピーするということです。そのため、元々あったパーソナルボキャブラリは上書きされてしまいます。

<Visual Basicでの例>

(呼びだし例) `SpeechCtrl1.AddPersonalPool("user1", "C:¥userfiles¥user_a")`

<Directorでの例>

(呼びだし例) `AddPersonalPool (Sprite x, "user1", "C:¥userfiles¥user_a")`

ソースディレクトリには次のような指定も可能です。

`ftp://ftp.myhomepage.ne.jp/userfiles/user_a`

`file:///C:/Documents%20and%20Settings/voice/userfiles/user_a`

CopyUserFile

long CopyUserFile(BSTR bstrUserID, BSTR bstrSrcDirectory)

ViaVoiceのユーザーファイルを指定されたディレクトリから指定されたユーザーIDのディレクトリにコピーします。ディレクトリが無い場合は自動生成します。コピー元としてはローカルディレクトリあるいは、FTPサーバーのディレクトリが指定できます。注意が必要なのは、元のディレクトリにはあるファイルをすべてコピーするということです。そのため、もともとあったユーザー情報はすべて上書きされてしまいます。

<Visual Basicでの例>

(呼びだし例) `SpeechCtrl1.AddPersonalPool("user1", "C:¥userfiles¥user_a")`

<Directorでの例>

(呼びだし例) `AddPersonalPool (Sprite x, "user1", "C:¥userfiles¥user_a")`

ソースディレクトリには次のような指定も可能です。

`ftp://ftp.myhomepage.ne.jp/userfiles/user_a`

`file:///C:/Documents%20and%20Settings/voice/userfiles/user_a`

4. 処理の流れ

Prezam Voice Toolを使用してアプリケーションを作成する際の処理の流れは、図1のようになります。
Directorでの開発で、Lingoスクリプトで記述する場合にも、次のような処理の流れになるように記述します。InitializeとTerminateに関しては、フレームの最初とフレームの最後で行うといいでしょう。

ボキャブラリーのロードは最初に行なっておいて、イネーブル、ディスエーブルにより切り替える方が、初期化時に時間はかかりますが、処理は早くなります。また、サウンドの出力を行なう場合には、一旦マイクをオフにしてから出力する必要があります。サウンド出力が終了して、再度認識を開始するときには、サウンドが停止してからマイクをオンにする必要があります。サウンドカードのリソースが全二重で使用できないカードの方が多いためです。

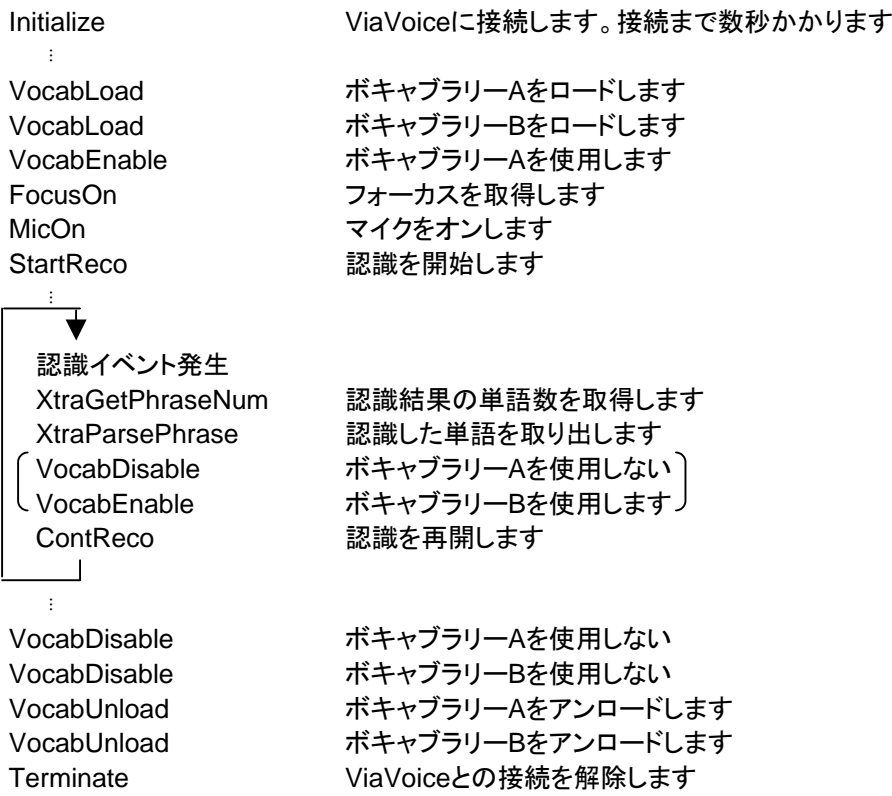


図 1

5. 認識イベントの処理

認識開始したあと、ユーザーがマイクに向かってしゃべると、それがグラマーに入っている語句の場合、認識イベントが発生します。認識イベントには、ユーザーが発声した語句がそのまま入ってきます。この語句を更に単語単位で返して欲しいという場合には、Directorの場合、以下のように二つのメソッドを呼び出すことによって取り出すことが可能となります。Visual Basicの場合には、ひとつのメソッドで取り出すことができます。

注)ディクテーション以外の認識イベントが発生した時には、エンジン認識開始を待っています。処理が終了して再度認識開始できるようになったら、ContRecoメソッドを呼び出して認識の再開をエンジンに促してください。認識単語によってボキャブラリの切替を次々に行えるようにするためこのようになっています。

```
on onPhraseRecognized Text
  Put XtraGetPhraseNum(Sprite x) into num
  Set x = []
  Set i = 0
  repeat while i < num
    Put XtraParsePhrase(Sprite x, i) into s
    x.append(s)
    i = i + 1
  end repeat
  :
  処理. . . .
  :
  ContReco(Sprite x)
end
```

6. フォーカスの移動処理

通常の処理では、ウィンドウがアクティブになったときにフォーカスを取得するのがいいでしょう。そこで、Directorの場合には、プロジェクターウィンドウがアクティブな時に音声フォーカスを取得し、ウィンドウがアクティブで無くなった時に音声フォーカスをリリースします。そのためには、次のように ActivateApplication ハンドラーで処理を行ないます。

プロジェクターがアクティブになったとき

```
on activateApplication
    FocusOn(true)
end activateApplication
```

プロジェクターがディアクティブになったとき

```
on deactivateApplication
    FocusOn(false)
end deactivateApplication
```


7. Visual Basic でのサンプル

sample フォルダにあるサンプルもご覧ください。他の ViaVoice を使用した音声認識アプリケーションと共存する場合は、自分がアクティブになった場合にマイクフォーカスを取得するようにします。sample フォルダにあるサンプルでは、タイマーイベントの中で自分がアクティブになると、マイクフォーカスを取得するようにしています。また、フォームのアクティブになったときも同じ処理をしています。

```
Private Sub Form_Load()
```

```
    SpeechCtrl1.Initialize
```

```
    rc = SpeechCtrl1.VocabLoad("ORDER", "d:¥order.fsg")
```

```
    rc = SpeechCtrl1.VocabEnable("ORDER")
```

```
    rc = SpeechCtrl1.FocusOn(True)
```

```
    rc = SpeechCtrl1.MicOn(True)
```

```
    SpeechCtrl1.StartReco
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
    SpeechCtrl1.Terminate
```

```
End Sub
```

```
Private Sub SpeechCtrl1_OnPhraseRecognized(ByVal pVal As String)
```

```
    Dim x As Integer
```

```
    Dim Phrase As Variant
```

```
    MsgBox pVal + " を認識した"
```

```
    Phrase = SpeechCtrl1.ParsePhrase
```

```
    For x = LBound(Phrase) To UBound(Phrase)
```

```
        List1.AddItem Phrase(x)
```

```
    Next
```

```
    Dim Annotation As Variant
```

```
    Annotation = SpeechCtrl1.PhraseAnnotation
```

```
    For x = LBound(Annotation) To UBound(Annotation)
```

```
        List1.AddItem Annotation(x)
```

```
    Next
```

SpeechCtrl1.ContReco

End Sub

8. グラマーファイルとプールファイルの作成について

グラマーファイルとプールファイルを作成するには、ViaVoice SDK が必要になります。ViaVoice SDK は IBM から入手してください。ダウンロードもできると思います。また、ViaVoice SDK は各国語共通になっています。

SDK の中で必要なものはグラマーコンパイラー、Dictionary Builder II というツールです。

グラマーコンパイラーは、テキストエディタで作成した認識させたい文章のグラマーソースファイルを ViaVoice が読める FSG 形式のファイルに変換するものです。

Dictionary Builder II は、グラマーで使用した単語を認識できるように単語と音素ラベルを対応させるためのプールファイルを作成するものです。

■グラマーファイル

詳しくはViaVoice SDKのプログラマーズガイドを参照していただくとして、ここでは簡単に説明します。

グラマーはBNF(Backus-Naur Form)により記述します。以下に簡単な例を示します。

「ファイルを開く」「ファイルを印刷する」「ファイルを閉じる」を認識してコマンドを実行するようなアプリケーションを作成する場合です。

```
<root> =   ファイル を 開く |  
           ファイル を 印刷 する |  
           ファイル を 閉じる.
```

処理の部分を<action>として別に分けた場合は以下のようにになります。

```
<root> =   ファイル を <action>.  
<action> = 開く | 印刷 する | 閉じる.
```

ファイルだけでなくメモでも認識するようにした場合は以下のようにになります。

```
<root> =   <object> を <action>.  
<object> = ファイル | メモ.  
<action> = 開く | 印刷 する | 閉じる.
```

コンパイルの方法

コンパイルはViaVoice SDKのフォルダーのToolsフォルダー以下に入っているvtbnfcコマンドを使用します。詳しくはViaVoice SDKのプログラマーズガイドおよびツール使用ガイドをご覧ください。

例 vtbnfc -cp 932 myvocabrary.bnf

コンパイルすると、myvocabrary.fsgと拡張子がfsglになったファイルが作成されます。これをボキャブラリファイルとして使用します。VocabLoad()でロードするファイルとして指定するわけです。

■プールファイル

プールの作成にはDictionary Builder IIを使用して単語の音素ラベルを登録していきます。

Dictionary Builder IIは、GUIのツールになっています。先ほど作成したmyvocabrary.fsgのプールの作成をする場合、メニューから[Spelling]–[Load]でこのfsgをロードします。

詳しくはViaVoice SDKのツール使用ガイドをご覧ください。

最終的には[Baseforms] – [Create Pool]でプールファイルを生成して使用します。

9. 使用上の注意

プールファイルを使用する場合には、プールファイルのインストール先に注意が必要です。

プールファイルは、発音と単語を結びつけるために ViaVoice により使用されるファイルです。このファイルをインストールするパスは、次のところにしてください。

1. Windows のレジストリーを参照します。
2. HKEY_LOCAL_MACHINE の下の以下のレジストリーキーを参照します。
SOFTWARE\IBM\VoiceType\Engine\Directories
3. ここに DataPath という項目がありますので、ここに書かれているパスにインストールします。%L という文字がパスに入っていますが、これは、ランゲージにより変わります。日本語の場合は [Ja_JP](#)、米国英語の場合、[En_US](#) となります。

VoiceTool3 より SetPool()メソッドが追加されています。これを使用すると、指定したプールファイルを上記のフォルダーにコピーします。便利なのでご利用ください。ただし、ViaVoice エンジンが停止している状態で無いとコピーしたプールファイルのデータが反映されません。これを使用しているサンプルが sample の Web フォルダに入っています。

10. リリースノート

2.0.0 Oct/30/2000

配布版リリース

2.1.0 Jul/01/2001

- ・Thresold プロパティの追加
- ・Rejection プロパティの追加
- ・register_メソッドの追加
- ・getVersion メソッドの追加
- ・PhraseVocabName メソッドの追加
- ・ディクテーションマクロを有効にしました。

2.1.0.2 Aug/08/2001

- ・リファレンスマニュアルを修正しました。
- ・一部処理を修正しました。

2.2.0.1 Aug/10/2001

- ・PhraseAnnotationString の追加
- ・xtraPhraseAnnotationString の追加

2.2.1.1 Aug/22/2001

- ・複数プロセスで使用時に音声フォーカスがうまく切り替わらないのを修正
- ・ドキュメントの VB のサンプルを修正とボキャブラリ、プールファイルについての記述を追加しました。

2.2.1.2 Nov/01/2001

- ・音声フォーカスの問題を修正。
- ・HTML による使用で不正終了する場合があるのを修正。
- ・Delphi のサンプルを追加しました。

新版 VoiceTool3

GUID を変更して今までの VoiceTool 2.0 とは別の ActiveX コントロールとしています。VoiceTool 2.0 と VoiceTool3 は共存できます。

3.3.1.1009 Jul/23/2002

- ・VV_Version プロパティを追加
- ・VVRtkClients_Launch メソッドを追加
- ・VVRtkClients_IsClientAvailable メソッドを追加
- ・VVRtkClients_IsClientRunning メソッドを追加
- ・ディクテーション時の数値の変換方法を変更
- ・SetPool メソッドを追加

- ・Terminate でトラップすることがあるのを修正
- ・AudioHost プロパティを追加し、カスタム Audio DLL に対応しました。
- ・SetAudioSourceParam メソッドを追加
- ・ファイル名指定の際のエスケープ文字(%20 など)に対応しました。

Version が 3.2 になりました。Version 3.2 からはパーソナル版と拡張版のふたつが存在します。

拡張版では、パーソナルボキャブラリおよびユーザーファイルのダウンロード機能および、パーソナルプールへの発音の追加機能が追加されます。

3.3.2.1029 Nov/05/2002

- ・GetUserList メソッドを追加
- ・XtraGetUserList メソッドを追加
- ・XtraGetUserListNum メソッドを追加
- ・1 度 Terminate すると Initialize してもイベントが発生しなかったバグを修正
- ・XtraGetUserList メソッドを追加
- ・GetViaVoicePath メソッドを追加
- ・ftp プロトコルを使ったボキャブラリやプールファイルのロードを追加
- ・SetHttpFtpMode メソッドを追加
- ・GetLastErrorDiscription メソッドを追加
- ・ボキャブラリがロードできないと、次の smapi 呼び出しから-4 のエラーになるため、トラップしていたのを修正
- ・AddPersonalPool メソッドを追加した(拡張版機能)
- ・Proxy を通さないと HTTP プロトコルでの FTP がうまく行かないのを修正した
- ・HTML でエラーが返ってきた場合に、エラーと認識するように修正した
- ・SetAudioSourceParam では Wave ファイルを何度も使用する可能性があるため、ユニークな一時ファイルを作成するように修正した
- ・CopyPersonalVocabulary メソッドを追加(拡張版機能)
- ・CopyUserFile メソッドを追加(拡張版機能)
- ・GetCurrentUserID メソッドを追加
- ・TerminateImmediately メソッドを追加
- ・Delay メソッドを追加
- ・AddPersonalPool メソッドで、日本語コードページ以外では baseform が必要なエラーを発生するようにした(拡張版機能)
- ・ディクテーションしているときに、ViaVoice Version10 でありえないイベントがくるために不正終了していたのを修正した
- ・CopyPersonalVocabulary で "*.adp" がコピーされていないバグを修正(拡張版機能)

3.3.2.1032 Dec/13/2002

- ・FTP によるパーソナルボキャブラリおよびユーザー情報のダウンロード時にダイア

ログを表示するようにした(拡張版機能)

- ・CopyPersonalVocabulary と CopyUserFile でプロセスチェックによる ViaVoice エンジン起動中かのチェックを入れた(拡張版機能)

3.3.2.1037 Jun/02/2003

- ・SetPool でプロセスチェックによる ViaVoice エンジン起動中かのチェックを入れた
- ・メッセージボックスをコンテナウィンドウを親ウィンドウにして出すようにした
- ・ディクテーションにひらがなモードを追加した
- ・ディクテーションのモード変更用メソッド SetDictMode を追加した
- ・HTTP プロトコルでのボキャブラリのロード時にボキャブラリのサイズによって途中までしかロードしていなかったバグを修正した

3.3.2.1041 Jan/15/2004

- ・速度の改善
- ・候補を取得するかどうかを設定できるようにした(速度改善のため) **デフォルトは取得する
- ・VocabLoadEx でも一回までは体験できるようにした。体験版のためのエラーコードも-30000 だったところを 40000 に修正した

3.3.2.1047 Aug/09/2004

- ・AppName と AudioSave プロパティをセットしても反映されていなかったのを修正した
- ・PlayWords メソッドを追加した
- ・「約 4 リットルに」などの場合、数値変換で tag 数と単語が合わなくなるため、修正した
- ・DiscardAudioData メソッドを追加した
- ・SaveWords メソッドを追加した

3.3.2.1048 May/06/2005

- ・IsReject メソッドを追加
- ・ディクテーションで、Reject プロパティが効いてなかったのを修正