

2005/04/05

Macromedia Director 音声認識 xtra xtraVVoice

現在のMacOSでは、タスクスケジューリングの制限によりDirectorとViaVoiceIにうまくタイムスライシングされません。そのため、MacOS版のXtraは正常に動作することができません。MacOS Xで動作するViaVoiceが無いいため、Mac版のxtraVVoiceの配布は中止しています。

目 次

目 次	2
extraVVoiceIについて	4
『ライセンス情報』	5
1. セットアップ	6
2. コマンドリファレンス	7
■レジスト/バージョン取得	7
register	7
getVersion	7
■インスタンスの新規作成	7
new	7
■初期化/終了	8
initialize	8
initializeEx	8
terminate	8
■動作設定	9
getRejection	9
putRejection	9
getThreshold	9
putThreshold	9
■グラマーボキャブラリ	10
vocabLoad	10
vocabLoadEx	10
vocabUnload	10
vocabEnable	11
vocabDisable	11
vocabAddWord	11
vocabRemoveWord	11
■マイク/マイクフォーカス	12

micOn	12
focusOn	12
■ 認識開始/終了	13
startReco	13
contReco	13
endReco	13
■ 認識結果の取得	14
getPhraseNum	14
parsePhrase	14
phraseAnnotation	14
parseAnnotationString	14
getPhraseAlternativesNum	15
phraseAlternatives	15
phraseVocabName	15
イベントハンドラ	16
phraseRecognized	16
changeMicState	16
changeFocusState	16
changeAudioState	17
4. 処理の流れ	18
5. 認識イベントの処理	19
6. フォーカスの移動処理	20
7. グラマーファイルとプールファイルの作成について	21
8. 使用上の注意	23
【Macでの使用上の注意】	23
【共通】	23
9. リリースノート	24

xtraVVoice について

この度は、xtraVVoice をご利用いただきまして、誠にありがとうございます。

この xtra は、IBM ViaVoice による音声認識を Macromedia Director で使用でき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 による音声認識を中心に作られていますが、Dictation による認識も一部サポートされています。

この xtra 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) MacOS と Windows の両方で動作するコンテンツを作成できます。

動作環境

Director 8.0J 以降のバージョン

Macintosh PowerPC MacOS 8.5.1, 8.6, 9.0

ViaVoice for Macintosh の動作環境に準拠します。

Windows95, Windows 98, Windows Me, Windows NT, Windows 2000

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、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。

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、『xtraVVoice』（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。

このソフトウェアの仕様、およびマニュアルに記載されている事柄は、将来予告なく変更されることがあります。

以上

1. セットアップ

Director がインストールされているフォルダの中の Xtras フォルダに、次のファイルをコピーしてください。

【Macintosh】

extraVVoice

【Windows】

extraVVoice.X32

2. コマンドリファレンス

■レジスト/バージョン取得

register

シンタックス	register("serial-number")
戻り値	シリアル番号が不正な場合はfalseを返します。
説明	xtraVVoiceをレジストします。シリアル番号を登録すると、複数のグラマーをロードできるなどフル機能がご利用できるようになります。
例	obj.register("xtvv-100-xxx-xxxxxx")

getVersion

シンタックス	getVersion()
戻り値	バージョン番号を返します。
説明	xtraVVoiceのバージョン番号を取得します。
例	obj.getVersion()

■インスタンスの新規作成

new

シンタックス	set obj = new(xtra "xtraVVoice")
説明	xtraVVoiceのインスタンスを新規に作成します。
例	set the cpuHogTicks = 6 set obj = new(xtra "xtraVVoice") if not objectP(obj) then exit obj.register("xtvv-100-xxx-xxxxxx") obj.getVersion() obj.Initialize() obj.VocabLoad("menu", "Macintosh HD:Sample:Grammar.fsg") obj.VocabEnable("menu") obj.FocusOn(true) obj.MicOn(true) obj.StartReco() obj.VocabDisable("menu") obj.VocabUnload("menu")

obj.Terminate()

set obj = 0

■ 初期化/終了

initialize

シンタックス

obj.Initialize

戻り値

SMAPIからのエラーコード。ただし、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。

説明

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。ムービーの最初で必ず一度だけ呼ぶ必要があります。

例

obj.Initialize()

initializeEx

シンタックス

obj.InitializeEx(userid, enrollid)

戻り値

SMAPIからのエラーコード。ただし、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。

説明

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。ムービーの最初で必ず一度だけ呼ぶ必要があります。

例

obj.InitializeEx("userid", "enrolled")

terminate

シンタックス

obj.Terminate()

説明

ViaVoice音声認識エンジンとの接続を解除します。終了時に一度だけ呼び出します。

例

obj.Terminate()

■動作設定

getRejection

シンタックス `obj.getRejection()`

戻り値 SMAPIからのエラーコード。

説明 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の現在の設定を取得します。

例 `obj.getRejection()`

putRejection

シンタックス `obj.putRejection(boolean)`

戻り値 SMAPIからのエラーコード。

説明 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。デフォルトはtrueです。

例 `obj.putRejection(false)`

getThreshold

シンタックス `obj.getThreshold()`

戻り値 SMAPIからのエラーコード。

説明 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の現在の値を取得します。戻り値は0～10の間です。

例 `obj.getThreshold()`

putThreshold

シンタックス `obj.putThreshold(num)`

戻り値 SMAPIからのエラーコード。

説明 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。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。

例 `obj.putThreshold(5)`

■ グラマーボキャブラリ

vocabLoad

シンタックス `obj.VocabLoad(vocabname, filename)`

戻り値 SMAPIからのエラーコード。

説明 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ファイル名を指定します。ファイル名はフルパスで指定しないと、カレントフォルダからのパスになります。また、このメソッドを複数呼び出して、複数のボキャブラリーをロードすることも可能です。Windows版では、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。

例 `obj.VocabLoad("VOCAB_NAME", "C:¥MyApp¥Grammar.fsg")`

vocabLoadEx

シンタックス `obj.VocabLoadEx(vocabname, filename, type)`

戻り値 SMAPIからのエラーコード。

説明 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ファイル名を指定します。ファイル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。Windows版では、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。

例 `obj.VocabLoadEx("VOCAB_NAME", "C:¥MyApp¥Grammar.fsg", 2)`

vocabUnload

シンタックス `obj.VocabUnload(vocabname)`

戻り値 SMAPIからのエラーコード。

説明 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。

例 `obj.VocabUnload("VOCAB_NAME")`

vocabEnable

シンタックス	<code>obj.VocabEnable(vocabname)</code>
戻り値	SMAPIからのエラーコード。
説明	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。
例	<code>obj.VocabEnable("VOCAB_NAME")</code>

vocabDisable

シンタックス	<code>obj.VocabDisable(vocabname)</code>
戻り値	SMAPIからのエラーコード。
説明	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。
例	<code>obj.VocabDisable("VOCAB_NAME")</code>

vocabAddWord

追加予定

シンタックス	<code>obj.VocabAddWord(vocabname, addwords[])</code>
戻り値	SMAPIからのエラーコード。
説明	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。
例	<code>obj.VocabAddWord("VOCAB_NAME",)</code>

vocabRemoveWord

追加予定

シンタックス	<code>obj.VocabRemoveWord(vocabname, removewords[])</code>
戻り値	SMAPIからのエラーコード。
説明	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。
例	<code>obj.VocabRemoveWord("VOCAB_NAME")</code>

■ マイク/マイクフォーカス

micOn

シンタックス	<code>obj.MicOn(<i>trueOrFalse</i>)</code>
戻り値	SMAPIからのエラーコード。
説明	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。

例	<code>obj.MicOn(true)</code>	マイクオン
	<code>obj.MicOn(false)</code>	マイクオフ

focusOn

シンタックス	<code>obj.FocusOn(<i>trueOrFalse</i>)</code>
戻り値	SMAPIからのエラーコード。
説明	マイクフォーカスを取得するか、他の音声認識プログラムに明渡すかを指定します。 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。

例	<code>obj.FocusOn(true)</code>	フォーカス取得
	<code>obj.FocusOn(false)</code>	フォーカスリリース

■認識開始/終了

startReco

シンタックス	<code>obj.StartReco()</code>
戻り値	SMAPIからのエラーコード。
説明	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。
例	<code>obj.StartReco()</code>

contReco

シンタックス	<code>obj.ContReco()</code>
戻り値	SMAPIからのエラーコード。
説明	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。
例	<code>obj.ContReco()</code>

endReco

シンタックス	<code>obj.EndReco()</code>
戻り値	SMAPIからのエラーコード。
説明	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。
例	<code>obj.EndReco()</code>

■認識結果の取得

getPhraseNum

シンタックス	<code>obj.GetPhraseNum()</code>
戻り値	単語数
説明	認識した語句の単語数を取得します。
例	<code>obj.GetPhraseNum()</code>

parsePhrase

シンタックス	<code>obj.ParsePhrase(word_index)</code>
戻り値	SMAPIからのエラーコード。
説明	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、GetPhraseNumで取得した値-1の値までです。たとえば、GetPhraseNum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。
例	<code>obj.ParsePhrase(word_index)</code>

phraseAnnotation

シンタックス	<code>obj.PhraseAnnotation(word_index)</code>
戻り値	SMAPIからのエラーコード。
説明	認識した単語の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、GetPhraseNumで取得した値-1の値までです。たとえば、GetPhraseNum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。
例	<code>obj.PhraseAnnotation(word_index)</code>

parseAnnotationString

シンタックス	<code>obj.ParseAnnotation(word_index)</code>
戻り値	SMAPIからのエラーコード。
説明	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。 アノテーションで文字列を指定した時にはこの関数で取得してください。 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、GetPhraseNumで取得した値-1の値までです。たとえば、GetPhraseNum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。
例	<code>obj.ParseAnnotation(word_index)</code>

getPhraseAlternativesNum

シンタックス `obj.GetPhraseAlternativesNum()`

戻り値 SMAPIからのエラーコード。

説明 認識した語句の候補単語の数を取得します。

例 `obj.GetPhraseAlternativesNum()`

phraseAlternatives

シンタックス `obj.PhraseAlternatives(word_index)`

戻り値 SMAPIからのエラーコード。

説明 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、`GetPhraseAlternativesNum`で取得した値-1の値までです。たとえば、`GetPhraseAlternativesNum`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。

例 `obj.GetPhraseAlternatives(word_index)`

phraseVocabName

シンタックス `obj.PhraseVocabName()`

戻り値 SMAPIからのエラーコード。

説明 認識した単語の属するボキャブラリ名を取得します。

例 `obj. PhraseVocabName ()`

イベントハンドラ

■ 語句認識イベント

phraseRecognized

シンタックス on PhraseRecognized *Text*

説明 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コマンドを呼び出して取得することになります。

例 on PhraseRecognized *Text*

end

■ マイクの状態変化イベント

changeMicState

シンタックス on ChangeMicState *state*

説明 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。

例 on ChangeMicState *state*

end

■ フォーカスの状態変化イベント

changeFocusState

シンタックス on ChangeFocusState *state*

説明 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。

例 on ChangeFocusState *state*

end

■オーディオの状態変化イベント

changeAudioState

シンタックス `on ChangeAudioState state`

説明 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。

例 `on ChangeAudioState state`

`end`

4. 処理の流れ

extraVVoice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。

Lingoスクリプトで記述する際に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よいでしょう。

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。サウンドリソースが全二重で使用できないカードがあるためです。

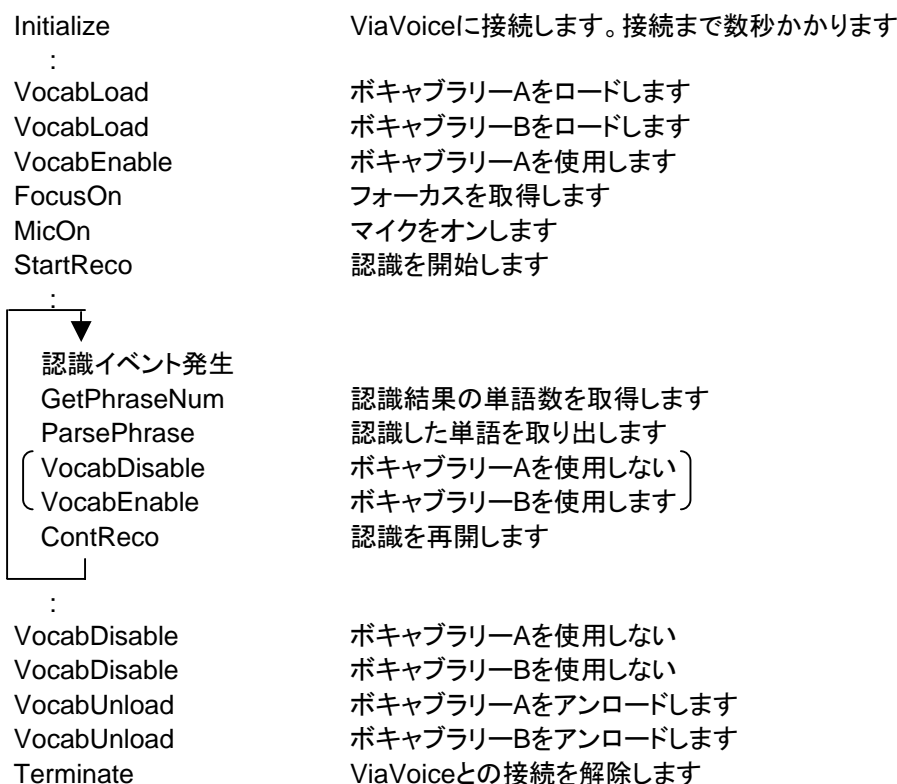


図 1

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、以下のように二つのコマン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。

注)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コマンドで認識の再開をエンジンに促してください。

```
on PhraseRecognized Text
  Put obj.GetPhraseNum() into num
  Set x = []
  Set i = 0
  repeat while i < num
    Put obj.ParsePhrase(i) into s
    x.append(s)
    i = i + 1
  end repeat
  :
  処理. . . .
  :
  obj.ContReco()
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、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
    obj.FocusOn(true)
end activateApplication
```

プロジェクターがインアクティブになったとき

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

7. 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 ホーム ページ の ボイス ら ん ど か ら ダ ウ ン ロ ード でき ます 。 現 在 の URL は <http://www6.software.ibm.com/dl/vv sdk/vv sdk-p> 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。いか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ようになります。

```
<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ようになります。

```
<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 fsg 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。

8. 使用上の注意

【Mac での使用上の注意】

ViaVoice に多くの処理をさせないとマイクから入力された音声を取りこぼしてエラーになります。そのため、Mac では Director の cpu 時間を ViaVoice へ多く明渡す必要があります。
cpuHogTicks でより多くの時間を他のプログラムに開放するようにセットしてください。

【共通】

グラマーの作成とコンパイル、プールファイルの作成については、IBM ViaVoice SDK に入っているドキュメントおよびツールを使用してください。IBM ViaVoice SDK は、IBM のサイトから無償でダウンロードできます。

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フォルダは、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。

なお、ViaVoice を使った音声認識ソフトの作り方については、有料にて開発サポートも行なっております。

9. リリースノート

1.0.0 Apr/15/2001

2.1.0 Aug/01/2001

- getThresold の追加
- putThresold の追加
- getRejection の追加
- putRejection の追加
- PhraseVocabName の追加
- ディクテーションマクロを有効にしました。

1.2.1.1 Aug/10/2001

- PhraseAnnotationString の追加
- 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
- ドキュメントにボキャブラリ、プールファイルについての記述を追加しました。

1.2.1.1 Nov/01/2001

- リファレンスマニュアルの誤りを訂正しました。

1.2.1.2 Apr/05/2005

- Director MX 2004 に対応しました。