

**P e t e完成版
カスタマイズマニュアル**

平成13年5月

アライド・ブレインズ株式会社

目次

| | |
|--|----|
| 1 . P e t e 動作環境設定方法..... | 1 |
| 1 - 1 . 書式..... | 1 |
| 1 - 2 . 用意されている変数名とその機能..... | 2 |
| 1 - 3 . 各変数詳細..... | 2 |
| 1 - 4 . P e t e がデフォルトで持っている値..... | 6 |
| 2 . ソフトキーボード設定方法..... | 7 |
| 2 - 1 . 書式..... | 7 |
| 2 - 2 . 色作成で指定できるアトリビュート名 (種別 = Color) | 8 |
| 2 - 3 . ソフトキーやグループ作成等で指定できるアトリビュート名 (種別 = Key) | 9 |
| 2 - 4 . マクロ記述..... | 12 |
| 2 - 5 . マクロ内コマンド記述 (Command) | 17 |
| 2 - 6 . その他..... | 20 |
| 2 - 7 . 特別なグループ (ウィンドウとして扱われる) | 22 |
| 3 . サンプル初級者編..... | 23 |
| 3 - 1 . 動作環境の設定..... | 23 |
| 3 - 2 . 色の設定..... | 23 |
| 3 - 3 . ソフトキーボード定義ファイルの作成 | 24 |
| 4 - 1 . キーを追加する..... | 35 |
| 4 - 2 . 単語予測ウィンドウを使用しない..... | 37 |
| 4 - 3 . 標準キーボードと相互に切り替えて使用するキーを追加する。 | 37 |

P e t eのカスタマイズマニュアル

P e t eの動作環境や画面構成はP e t eプログラムが置いてあるディレクトリと同じところに置いてあるファイルの内容を書き換えることによって変更することができます。

1 . P e t e動作環境設定方法

P e t e動作環境設定とは、スキャンのスピード変更やソフトキーボード画面の大きさの変更等P e t eが持っている機能のオプションを設定することです。何も手を加えていない場合はデフォルトの設定になっていますので、使ってみて字を大きくしたい等の要望がでてきたとき、プログラムを変えることなく、この設定でP e t eの動作を変更することができます。

P e t eは起動したときのみこの設定を読み込むので、起動中に設定を変えても動作は変わりません。設定を変えた場合は必ずP e t eを終了してから再度起動してください。

設定するためのファイル名はP e t eプログラムと同じディレクトリに置いてある「settings.txt」というファイルで、修正後も必ず同じディレクトリにおかなくてはなりません。

1 - 1 . 書式

(1) 基本形 (必ず 1 行に書く)

変数名 = 値

(2) 値について

「値」には以下の3タイプが指定でき、変数名によってタイプが一意に決まります。

1) 数値型。

2) 2値論理型。「True/False」または「0/1」でその機能を使用するか否かを指定します。

3) 文字列型。文字列は必ず ” ” で括らなければなりません。

(3) サンプル

```
# 表示関連
#
# FrameWidth=2
# FontBold=False
# MainWindowSize=0 (auto)
# MediumFontSize=0 (auto)
# MetricsMode=1
FontBold=True
MetricsMode=2
```

注) ‘ # ’ で始まっている行はコメント行です。

1 - 2 . 用意されている変数名とその機能

| 変数名 | 機能 |
|-----------------|---------------------------|
| TimerInterval | スキャンのスピードを指定 |
| TimerWait | スキャン時グループの先頭で余分に待機する時間を指定 |
| TimerFlag | Pete起動時にスキャンで動作する/しないを指定 |
| TimerSameGroup | スキャン時同一グループ内の繰り返し回数を指定 |
| FrameWidth | ソフトキーのフレームの太さを指定 |
| FontBold | 文字を太く表示する/しないを指定 |
| MainWindowSize | ウインドウの縦の大きさを指定 |
| MediumFontSize | 標準のFontの大きさを指定 |
| MetricsMode | キーレイアウトの座標系のタイプを指定 |
| KeyCfgFile | レイアウトを記述するファイル名を指定 |
| ColorCfgFile | 色定義を記述するファイル名を指定 |
| LogFile | 詳細ログを記録するファイル名を指定 |
| CounterFile | カウントを記録するファイル名を指定 |
| SettingFile | この設定ファイルの名前 (未使用) |
| LoggingMode | 詳細ログをとるかをとらないかを指定 |
| CounterMode | ソフトキーストロック数をとるかをとらないかを指定 |
| MaxCandidates | 1度に表示する候補の最大数を指定 |
| PressKey | どのハードキーをスキャン時の選択キーにするのかを指 |
| MouseOnScan | スキャン時にマウス入力を許可するかを指定 |
| ProcessOnQuit | 終了時に実行するコマンドを指定 |
| KillOnQuit | 終了時に終了させるコマンドを指定 |
| KillProcessWait | 外部プロセスを終了の待ち時間を指定 |
| BeepMode | スキャンの移動をビーブ音で知らせるか否かを指定 |
| ForceMouseMove | Pete起動時のマウス制御を指定 |

1 - 3 . 各変数詳細

(1) TimerInterval (数値型)

TimerInterval=6

スキャンモードの時、スキャンが移動するスピードを0.1秒単位(0.1秒が1に相当)で指定します。スピードが速いようでしたら1つつ大きくして、遅く感じたら1つつ減らして設定を調整してください。

(2) TimerWait (数値型)

TimerWait=8

スキャンモードでソフトキーのグループにスキャンが移った時、その先頭にあるソフトキーにどれだけフォーカスが留まるか(スキャンが移動しないか)を0.1秒単位(0.1秒が1に相当)で指定します。この変数に数値を与えておかないと、グループの先頭ソフトキーが選択しづらくなります。また、先頭ソフトキーにフォーカスが留まる時間は“TimerInterval”で指定した値との合計値になります。

(3) TimerFlag (2 値論理型)

TimerFlag=False 又は TimerFlag=0

Peteを立ち上げたときからすぐにスキャンモードで動作するかどうかを指定します。ス

キャンモードでない場合、P e t e 立ち上げ時「Scan」キーにフォーカスがありますので、1 回スキャン確定キー（デフォルトで“無変換”キー）を選択すればスキャンが始まります。また、P e t e が起動している時でも「Scan」キーでスキャンモードのON / OFF が動的に変更可能なようになっています。

(4) TimerSameGroup (数値型)

TimerSameGroup=3

スキャンモードで動作している時、同一グループ内のソフトキー又はグループを3周（デフォルト動作）すると自動的にゾーン選択に戻るようになっていきます。これは間違っ隣グループを選択したときなど、また最初からやり直したいときのために用意した機能です。スキャンが3周するのが待ちきれないときなどはより小さい値を、いつもやり過ぎて3周では目的のソフトキーを選びにくいという方はより大きな値を設定してください。（0は設定不可）

(5) FrameWidth (数値型)

FrameWidth=2

各グループ及びソフトキーの枠表示の線の太さをドット数で指定します。スキャンがどこを周っているか見難い時は太くすると見やすくなります。

(6) FontBold (2 値論理型)

FontBold=False 又は FontBold=0

ソフトキーボードの刻印文字が見難い時、その刻印文字を太くすることができます。但し、選択肢は“細い”か“太い”(0 / 1 型) の2つしかありません。つまり、数値による文字の太さは指定できません。

(7) MainWindowSize (数値型)

MainWindowSize=0

P e t e 画面の縦方向の高さをドット単位で指定します。デフォルト（値が0）ではお使いのパソコンの画面サイズの1 / 3がP e t e の画面の高さになっています。

(8) MediumFontSize (数値型)

MediumFontSize=0

ソフトキーの刻印文字や予測ウィンドウの候補文字のサイズをフォントサイズで指定します。デフォルト（値が0）ではP e t e の画面サイズに応じて自動計算されたフォントを使用します。

(9) MetricsMode (数値型)

MetricsMode=1

ソフトキーやグループを「pete_key.cfg」（デフォルト）というファイルで自由にレイアウトすることが出来ますが、そのファイルに記述するソフトキーの表示配置座標を相対系で指定するか絶対系で指定するかを1（相対系）/ 2（絶対系）で指定します。相対系は1つのグループの中を100000×100000の仮想座標で指定し、絶対系はパソコンの画面全体を100000×100000とみなして計算する方式です。（、版は相対系、完成版は絶対系で作ってあります。）

- (1 0) KeyCfgFile (文字列型)
KeyCfgFile= " Pete_key.cfg "
P e t e ソフトキーボードを定義するファイル名を指定します。デフォルトでは「pete_key.cfg」になっていますが、独自の定義ファイルを新たに作成しそれを呼び出すこともできます。また、定義ファイル内のキー定義でコマンド (LoadCfg) を利用すれば動的に (P e t e が立ち上がったままで) 定義ファイルを切り替えることも可能です。
- (1 1) ColorCfgFile (文字列型)
ColorCfgFile= " pete_color.cfg "
P e t e ソフトキーボードが使用する色パレットを定義するファイル名を指定します。デフォルトでは「pete_key.cfg ファイル内に記述されている」ことになっていますが、その色定義部分を別のファイルに切り出して作成し、それを呼び出すことができます。(標準で提供しているものは「pete_color.cfg」というファイルに取り出した状態になっている。)
- (1 2) LogFile (文字列型)
LogFile= " LogFile.txt "
P e t e はソフトキー操作で何が押されたかなどの情報を記録 (ログ) しますが、そのログを取るファイル名を指定します。但し、LoggingMode が T r u e になっていないとログはとられません。
- (1 3) CounterFile (文字列型)
CounterFile= " counter.txt "
P e t e はソフトキーが何回押されたかの情報を記録 (ログ) しますが、そのログを取るファイル名を指定します。但し、CounterMode が T r u e になっていないとログはとれません。
- (1 4) SettingFile (文字列型)
SettingFile= " settings.txt "
環境設定ファイル名を指定します。デフォルトは " settings.txt " です。しかし、このファイル名は既にプログラムに組み込まれていますので、現在は指定しても無効です。
- (1 5) LoggingMode (2 値論理型)
LoggingMode=False 又は LoggingMode=0
P e t e はソフトキー操作で何が押されたかなどの情報を記録 (ログ) することができますが、それを行うか否かを指定します。(Pete の性能を精査されるかのための機能)
- (1 6) CounterMode (2 値論理型)
CounterMode=False 又は CounterMode=0
P e t e はソフトキーを何回押されたかの情報を記録 (ログ) することができますが、それを行うか否かを指定します。(Pete の性能を精査されるかのための機能)
- (1 7) MaxCandidates (数値型)
MaxCandidates=10
予測ウィンドウに表示する候補の最大数を指定します。
- (1 8) PressKey (文字列型)

PressKey= " NonConvert "

スキャンモードの時、キーボード（実際のキーボード）の特定のハードキーボードのキーからの入力でスイッチ入力を代用できます。その代用キーの割り当てを指定します。デフォルトは「無変換」キーになっています。また、指定できるキーの種類は“ 2 - 4の(5)Directマクロ” で使用する「タグ名」と同じものです。

(1 9) MouseOnScan (2 値論理型)

MouseOnScan=False 又は MouseOnScan=0

スキャンモード動作中に Pete 画面内のマウス操作を有効にするか否かを指定します。

(2 0) ProcessOnQuit (文字列型)

ProcessOnQuit= " "

ここにアプリケーションの起動コマンドを記述しておく、P e t e が終了した時、そのアプリケーションを起動することが出来ます。

(2 1) KillOnQuit (文字列型)

KillOnQuit= " "

ここにアプリケーションの起動コマンドを記述しておく、P e t e が終了した時、そのアプリケーションを終了させることが出来ます。また、ProcessOnQuit が実行される前に実行されます。

(2 2) KillProcessWait (数値型)

KillProcessWait=4

KillOnQuit が指定されている時、そのアプリケーションに強制終了メッセージ送信後何もしないで待つ時間（そのアプリケーションの後処理時間）を指定します。

(2 3) BeepMode (2 値論理型)

BeepMode=False 又は BeepMode=0

スキャンモード動作時、フォーカスが移動するごとにビーブ音を鳴らすか否かを指定します。

(2 4) ForceMouseMove (数値型)

ForceMouseMove=0

P e t e 起動時のみマウスポインタの位置を制御できます。指定できる数値は0～2でそれぞれ意味は、「何もしない」(0)「P e t e 画面の外に移動する。」(1)「P e t e 画面内に移動する」(2)です。これは、スキャンで使われるかたで、マウスポインタが入力の邪魔になったりしたときなどに使えます。

1 - 4 . P e t eがデフォルトで持っている値

| 変数名 | デフォルト値 | 意味 |
|-----------------|----------------|--------------------------|
| TimerInterval | 6 | スキャンスピードが600ms |
| TimerWait | 8 | グループ最初のキーに留まる時間 |
| TimerFlag | False | Pete立ち上げ時スキャンしない |
| TimerSameGroup | 3 | スキャンが同じグループ内を3周したらゾーン選択へ |
| FrameWidth | 2 | フレームの太さが2ドット |
| FontBold | False | 刻印文字の太さは標準 |
| MainWindowSize | 0 | Pete画面の大きさは画面の1/3 |
| MediumFontSize | 0 | 刻印文字のフォントサイズは自動計 |
| MetricsMode | 1 | 相対系座標でレイアウト |
| KeyCfgFile | "pete_key.cfg" | キーレイアウトはこのファイルで |
| ColorCfgFile | " " | 色定義はKeyCfgFileのファイル内 |
| LogFile | "LogFile.txt" | ログはこのファイルに記録 |
| CounterFile | "Counter.txt" | キーの押下回数はこのファイルに記録 |
| SettingFile | "Settings.txt" | 未使用 |
| LoggingMode | False | ログは記録しない |
| CounterMode | False | キーの押下回数は記録しない |
| MaxCandidates | 10 | 候補は最大10個表示させる。 |
| PressKey | NonConvert | スキャン時の確定キーは無変換 |
| MouseOnScan | True | スキャン時でもマウスを有効 |
| ProcessOnQuit | " " | Pete終了時起動するAPなし |
| KillOnQuit | " " | Pete終了時終了させるAPなし |
| KillProcessWait | 4 | 無効 (KillOnQuitが指定なし) |
| BeepMode | 0 | スキャン時移動時のビーブなし |
| ForceMouseMove | 0 | Pete立ち上がり時マウス制御なし |

2 . ソフトキーボード設定方法

ソフトキーボードの設定とは標準で提供しているソフトキーボードの配置を変更したいときや新しいソフトキーを追加したりしたいとき、更にはまったく新しいソフトキーボードを作成したいときに必要となります。本章は、そのファイルの記述ルールを言語的に説明しているものですので、まず標準で提供しているファイルや後章で挙げるサンプルを参照され、本章の説明は、わからないときなどに辞書的に使用されることをお勧めします。

また、設定するためのファイルは、“Settings.txt”内で指定するか以下で述べる定義ファイルのロード命令で指定してください。デフォルトでは「pete_key.cfg」というファイル名になっています。全てのソフトキーボードの設定のためのファイルは必ずP e t eプログラムと同じディレクトリに置いておかななくてはなりません。

キーボードの配色も自由に設定できますが、標準で提供しているソフトキーボード設定ファイル（Pete_key.cfg）では配色を定義せず、別ファイル（PeteColor.cfg）で定義しています。

2 - 1. 書式

(1) 基本形

種別 (番号) { アトリビュート }

種別 : Color、Key があり、Color で色を、Key でソフトキーやグループ等を生成します。

番号 : ID と呼び、同一種別内では同じ番号を 2 つ以上指定できません。

範囲は 1 ~ 2 の 3 2 乗 - 1

アトリビュート : 種別により記述できるアトリビュート名が異なります。

注) ‘ # ’ で始まっている行はコメント文です。

(2) 種別 Color の役割

ソフトキー、グループ、タブ、予測ウィンドウの候補、更には「次候補」「前候補」「IME 候補」のボタンで使用する色のパレットを生成するための種別です。

(3) 種別 Key の役割

ソフトキー、グループ、タブそしてコメントを生成するための種別です。

(4) 基本形の記述フォーマット

Key(ID){ Attributes=value }

Color(ID){ Attributes=value }

注) 種別 key でソフトキー、グループ、タブ、コメントを生成する。

(5) アトリビュート内書式

アトリビュート名 = 値 (Attributes=value)

値に記入する形式はアトリビュート名により異なります。

2 - 2 . 色作成で指定できるアトリビュート名 (種別 = Color)

- (1) 全アトリビュートに共通の “ 値 ” について
値には色を「RRGGBB」の形式で指定します。RRは赤色の強さを16進2桁、GGは緑色の強さを16進2桁、BBは青色の強さを16進2桁で表したものです。
- (2) Normalアトリビュート
Normal = RRGGBB
通常状態の時の背景色を指定します。
- (3) Focusアトリビュート
Focus = RRGGBB
フォーカスを得た時の背景色を指定します。
- (4) Selectアトリビュート
Select = RRGGBB
選択された時の背景色を指定します。
- (5) fNormalアトリビュート
fNormal = RRGGBB
通常の状態の時のフレーム色を指定します。
- (6) fFocusアトリビュート
fFocus = RRGGBB
フォーカスを得た時のフレーム色を指定します。
- (7) fSelect
fSelect = RRGGBB
選択された時のフレーム色を指定します。
- (8) tNormalアトリビュート
tNormal = RRGGBB
通常の状態の時の文字色を指定します。
- (9) tFocusアトリビュート
tFocus = RRGGBB
フォーカスを得た時の文字色を指定します。
- (10) tSelect
tSelect = RRGGBB
選択された時の文字色を指定します。

2 - 3 . ソフトキーやグループ作成等で指定できるアトリビュート名 (種別 = Key)

(1) Typeアトリビュート (省略不可)

生成するもののタイプを指定するアトリビュートです。代入する値でソフトキー、グループ、コメント、タブのどれを生成するか指定します。

書式 : Type=数値

数値 = 1 : グループ

数値 = 3 : ソフトキー

数値 = 6 : コメント

数値 = 7 : タブグループ

数値 = 5 : タブ

(注) タブとは Windows のダイアログで表示されるタブと同じイメージで、Pete 内でもタブのタブを選択することでソフトキーボードを切り替えることができる機能のことで、標準でサポートしているソフトキーボード設定ファイルでは使用していません。Pete 版のソフトキーボードで使用しています。

(2) Parentアトリビュート (省略不可)

生成するものがどのグループに所属するか親グループIDを指定します。Pete 画面自身はID = 0で定義していますので、その直下にあるグループ (Pete 内のウィンドウ) は0を指定します。Pete が標準で提供しているファイルではID = 1をPete 画面自身と同じ大きさのグループとして定義 (ID = 1) していますので、カスタマイズ時はこのグループ下に生成することをお勧めします。

この種別はスキャンのために設定するもので、Pete プログラム自身を頂点としたツリー構造を構成します。スキャン時各階層 (グループ) をスキャンが巡回し、ユーザが何かを選んだらその下の階層に移行してユーザの次の選択を待ちます。選んだものがソフトキーならそのキーの機能を実行した後頂点に戻ってスキャンします。同一階層内でスキャンする順番はソフトキー設定ファイル内に記述した順番になります。

また、あるソフトキーが選ばれたとき次のスキャンをツリーの頂点以外に行かせたいときは、その行かせたいところの階層 (グループ) をすることになりますが、そのとき本IDを使用します。

書式 : Parent=ID

(3) Nameアトリビュート (省略不可)

Typeがソフトキーならばそのキーの刻印名になり、グループならばグループ名、タブグループ、タブならば各々タブグループ名、タブ名、コメントならばコメント文になります。更にグループ名の場合、Pete プログラムによって予約されている名前 (2 - 7参照) を使用すると特別なウィンドウとして扱われ、2つ以上のグループに同じ名前を使用できません。予約されていない名前に関しては何度使用しても構いません。

書式 : Name="文字列"

(4) X, Y (省略不可)

自分が所属するグループの大きさを10000 x 10000 (相対系座標) 100000

×100000 (絶対系座標)の座標系とみなし、その表示位置(左上が0,0)のX,Y座標を指定します。

座標系の選択は動作環境設定で行います。(1-3-(9)を参照)

相対系:

書式 : X = 0 ~ 10000

書式 : Y = 0 ~ 10000

絶対系:

書式 : X = 0 ~ 100000

書式 : Y = 0 ~ 100000

(5) CX, CY (省略不可)

自分が所属するグループの大きさを100000×100000の座標系(相対系座標)PCの画面全体の大きさを100000×100000の座標系(絶対系座標)とみなし、その表示サイズ(大きさ)を指定します。

書式 : CX = 0 ~ 100000

書式 : CY = 0 ~ 100000

(6) Colorset (省略不可)

そのキー、グループ、タブやコメントが使用する色を色指定で定義した色パターン(ID)から選んで指定します。

書式 : Colorset = Color ID

(7) onSelect

キー、グループやタブが選択された状態(マウスで言えば、左ボタンを押した状態)になった時、Peteがどのような振る舞い(処理)をすべきかをマクロで指定します。マクロとはPete自身が持っているPeteへの命令のことです。

書式 : onSelect="マクロ"

マクロの書式については後節2-4参照。

現在標準で提供しているソフトキーボード設定ファイルでは未使用です。

(8) onFocus

キー、グループやタブにフォーカスが移った状態になった時、Peteがどのような振る舞いをすべきかをマクロで指定します。

書式 : onFocus="マクロ"

マクロの書式については後節2-4参照。

(9) onUnselect

キー、グループやタブが選択された状態(マウスで言えば、左ボタンを押した状態)から別のキー、グループまたはタブに選択が移った時、Peteがどのような振る舞いをすべきかをマクロで指定します。

書式 : onUnselect="マクロ"

マクロの書式については後節2-4参照。

現Peteでは未使用です。

(10) onUnfocus

キー、グループやタブがフォーカスを得ていた状態から別のキーまたはグループにフォーカスが移った時、Peteがどのような振る舞いをすべきかをマクロで指定します。

書式 : onUnfocus="マクロ"

マクロの書式については後節 2 - 4 参照。

(11) onSelected

キー、グループやタブが選択された (マウスで言えば、同じキーの上で左ボタンを押して放した状態) 時、Peteがどのような振る舞いをすべきかをマクロで指定します。

書式 : onUnselect="マクロ"

マクロの書式については後節 2 - 4 参照。

(12) Visible

Pete は使用する全てのキー、グループ、タブグループ、タブ、コメントをソフトキーボード設定ファイルに記述しておかなくてはなりません、Pete が起動したとき最初から表示したいものか否かを指定します。

設定するときは原則 “ 1 ” にして、Pete 起動時には表示したくないグループのトップのグループのみを “ 0 ” にします。この属性を動的 (Pete が動いている最中) に切り替えるための命令を用意しています。この命令をソフトキーのアトリビュート名 (7) ~ (1 1) のマクロに記述することによって、ソフトキー選択で別のソフトキーボードに切り替えることが可能になります。

Visible=1 or 0

2 - 4 . マクロ記述

(1) 概要

マクロとは Pete プログラム自身を持っている機能をソフトキーボード設定ファイルから呼び出せるようにした Pete への命令群のことです。標準で提供しているソフトキーボードは幾つかのマクロを組み合わせて使用し、Pete としての機能を実現しています。

マクロ記述欄にはスペースを区切り文字として、複数のマクロを記述することができます。

以下の書式はその 1 つの基本形を示しています。

書式：マクロ名 (引数)

引数はマクロの種類によって異なります。また、マクロ名とその動作概要は以下の通りとなっています。

| マクロ名 | 動作 |
|--------------------|--------------------------------------|
| Translate | 変換後の文字列をアプリケーションに送信後、学習を行う。 |
| TranslateP | 同上 (フレーズ辞書用) |
| TranslateW | 同上 (単語辞書用) |
| TranslateD | 同上 (読み自身用) |
| TranslateI | 同上 (I M E 引用用) |
| Ptranslate | アプリケーションに操作コード送信後、学習を行う。 |
| DirectPhrase | 文字列を直接アプリケーションに送信する。 |
| AddRead | 読みの文字列に追加する。 |
| Direct | 登録されているキーを直接押したのと同じ行為になる。 |
| Lock | 指定されたキーコードをリピート送信する。 |
| LockSlow | Lock 機能で、リピートをゆっくり行う。 |
| LockQuick | Lock 機能で、リピートを速く行う。 |
| Process | 他プロセスを起動する。 |
| Kill | 他プロセスを強制終了させる。 |
| Wait | 時間待ちする。 |
| ReleaseSpecialKeys | Shift, Ctrl, Alt キーロックの解除を行う。 |
| Command | 登録されているコマンドを実行する。 |
| LoadCfgC | 予測ウィンドウクリア後、コンフィグレーションファイルの再読み込みを行う。 |
| LoadCfgNc | コンフィグレーションファイルの再読み込みを行う。 |
| SetFocus | スキャンが次に移るソフトキーを指定する。 |
| GetAllCandidates | 予測ウィンドウに全候補を表示する。 |
| GetImeCandidates | 予測ウィンドウに I M E 候補のみを表示する。 |

(2) Translate、TranslateP、TranslateW、TranslateD、TranslateI

書式：TranslateX (変換後の文字列 + ¥n + 読みの文字列)

単語予測ウィンドウに表示されている候補キーが使用するマクロで、通常のソフトキーのマクロとしては使用しません。

(3) PTranslate

書式：PTranslate ()

操作予測ウィンドウに表示されている候補キーが使用するマクロで、通常のソフトキーのマクロとしては使用しません。

(4) DirectPhrase

書式：DirectPhrase(出力文字列)

漢字コード変換機能を使わず(「読み」に入らない)指定された「出力文字列」を直接アプリケーションに送信するマクロです。

(5) AddRead

書式：AddRead(文字)

「文字」で指定した文字を漢字変換対象に追加し、フレーズ辞書・単語辞書・IME辞書を参照し、予測ウィンドウに単語予測候補を表示するマクロです。入力した直接の「ひらがな」と「カタカナ」は候補の先頭に必ず現れます。

また、直前に入力した「読み」のひらがな1文字を濁点(「ばびぶべぼ」等)や半濁点(「ばびぶべぼ」)にしたいときは

AddRead(`) ==> 濁点

AddRead(°) ==> 半濁点

のように記述します。

濁点や半濁点のないひらがなの場合は濁点・半濁点記号が入力されます。

(6) Direct

書式：Direct(タグ名)

「タグ名」で指定したハードウェアキーを直接アプリケーションに送る(アプリケーションにWindowsメッセージを送信する)マクロです。例えば、エスケープキー、シフトキー、マウスボタン等日本語入力キーではないキーボード(マウス)のキーをPeteにも入力させたいとき使用します。

タグ名の種類と「入力されるキー」の対応表は以下の通りとなっており、それぞれのキーの役割についてはWindowsマニュアルのWindowsキーイベントドキュメントを参照してください。

| タグ名 | 入力されるキー | タグ名 | 入力されるキー | タグ名 | 入力されるキー |
|----------|-------------|-----------|-------------|------------|------------|
| LButton | マウス左ボタン | Num0 | “ 0 ” キー | NumLock | NumLock キー |
| RButton | マウス右ボタン | Num1 | “ 1 ” キー | ScrollLock | VK_SCROLL |
| Break | Break キー | Num2 | “ 2 ” キー | A | 'A' キー |
| MButton | マウス中ボタン | Num3 | “ 3 ” キー | B | 'B' キー |
| Back | バックスペースキー | Num4 | “ 4 ” キー | C | 'C' キー |
| Tab | タブキー | Num5 | “ 5 ” キー | D | 'D' キー |
| Clear | クリアキー | Num6 | “ 6 ” キー | E | 'E' キー |
| Enter | エンターキー | Num7 | “ 7 ” キー | F | 'F' キー |
| Shift | シフトキー | Num8 | “ 8 ” キー | G | 'G' キー |
| Ctrl | コントロールキー | Num9 | “ 9 ” キー | H | 'H' キー |
| Alt | アルトキー | Multiply | Multiply キー | I | 'I' キー |
| Pause | Pause キー | Plus | “ + ” キー | J | 'J' キー |
| CapsLock | CapsLock キー | Separator | “ * ” キー | K | 'K' キー |
| Kana | カナキー | Minus | “ - ” キー | L | 'L' キー |

| | | | | | |
|-------------|---------------|---------|--------------------|---|-------|
| Esc | エスケープキー | Decimal | “ Del ” キー | M | 'M'キー |
| Convert | 変換キー | Divide | “ / ” キー-VK_DIVIDE | N | 'N'キー |
| NonConvert | 無変換キー | F1 | ファンクション 1 | O | 'O'キー |
| Accept | Accept キー | F2 | ファンクション 2 | P | 'P'キー |
| ModeChange | ModeChange キー | F3 | ファンクション 3 | Q | 'Q'キー |
| Space | スペースキー | F4 | ファンクション 4 | R | 'R'キー |
| PageUp | PageUp キー | F5 | ファンクション 5 | S | 'S'キー |
| PageDown | PageDown キー | F6 | ファンクション 6 | T | 'T'キー |
| End | エンドキー | F7 | ファンクション 7 | U | 'U'キー |
| Home | ホームキー | F8 | ファンクション 8 | V | 'V'キー |
| Left | “ ” キー | F9 | ファンクション 9 | W | 'W'キー |
| Up | “ ” キー | F10 | ファンクション 10 | X | 'X'キー |
| Right | “ ” キー | F11 | ファンクション 11 | Y | 'Y'キー |
| Down | “ ” キー | F12 | ファンクション 12 | Z | 'Z'キー |
| Select | Select キー | F13 | ファンクション 13 | 0 | '0'キー |
| Print | Print キー | F14 | ファンクション 14 | 1 | '1'キー |
| Execute | Execute | F15 | ファンクション 15 | 2 | '2'キー |
| PrintScreen | 画面転写キー | F16 | ファンクション 16 | 3 | '3'キー |
| Ins | 挿入キー | F17 | ファンクション 17 | 4 | '4'キー |
| Del | 1文字削除キー | F18 | ファンクション 18 | 5 | '5'キー |
| Help | Help キー | F19 | ファンクション 19 | 6 | '6'キー |
| Lwindows | 左の Windows キー | F20 | ファンクション 20 | 7 | '7'キー |
| Rwindows | 右の Windows キー | F21 | ファンクション 21 | 8 | '8'キー |
| Apps | Apps キー | F22 | ファンクション 22 | 9 | '9'キー |
| | | F23 | ファンクション 23 | | |
| | | F24 | ファンクション 24 | | |

上記以外に Windows とは関係ない Pete独自の機能として「BS」というタグがあります。これはタグ名「Back」とは違い、Direct(BS)と指定すると、Pete内に“読み”の文字があれば一番最後に入力した1文字を、なければアプリケーション側の文字カーソルの手前にある文字を削除します。後者の機能がタグ名「Back」に相当します。

(7) Lock

書式：Lock(タグ名)

“タグ名”で指定されたキーコードを繰り返しアプリケーションに送信します。“タグ名”の内容は(6)のDirectと同じです。また、このマクロ使ったソフトキーを選択するとフォーカスが外れるまで同じコードを一定間隔でアプリケーションに送信し続けます。つまり、キーボードのリピート機能に相当するものです。

スキャンモードでソフトキーが選択されると、スキャンがそのソフトキーに留まり、再度選択されたときスキャンが再開(フォーカスが次に移る)され、その間同じコードが繰り返し送られています。マウスならばそのソフトキーを選択してからもう一度クリックされるか、マウスポインタを外すまで、リピートします。

これはカーソル移動キーなどに用いると便利です。また、標準で提供しているソフトキーボードではアプリケーションのメニューバーをスキャンさせるのにも使用しています。

(8) LockSlow

書式 : LockSlow (タグ名)

機能は (7) の Lock と同じですが、リピート時間がゆっくりになります。

(9) LockQuick

書式 : LockQuick (タグ名)

機能は (7) の Lock と同じですが、リピート時間が速くなります。

(10) Process

書式 : Process(プログラム名)

“プログラム名”で指定したプログラムを起動するマクロです。プログラム名はそのプログラムが置かれているディレクトリも含めて絶対パスで指定しないと動作しません。

(11) Kill

書式 : Kill(プロセス名)

“プロセス名”で指定したプロセスを強制終了させるマクロです。プロセス名は Windows が管理している名前です。Ctrl+Alt+Del でプロセス名一覧が出ますが、その一覧に出てきている名前を指定してください。もし、該当するプロセスが存在しない場合は何もしません。

(12) Wait

書式 : Wait(時間)

一定時間何もしないで待つためのマクロです。引数の“時間”は 1 0 0 m s 単位で指定します。このコマンドは (11) の Kill マクロ実行後に相手プロセスの終了処理待ち合わせなどのタイミングを計りたいとき使われます。また、単独でも使えますが、Kill マクロとあわせて使うのが普通です。

(13) ReleaseSpecialKeys

書式 : ReleaseSpecialKeys()

Shift, Ctrl, Alt の 3 つのキーは Windows 内部でキーの状態を保持している特殊なキーです。その保持状態を全部リセットするマクロです。

標準で提供しているソフトキーボード設定ファイルでは「Win 操作画面」で Shift, Ctrl, Alt のソフトキーを選択後現れるソフトキーボードで、キャンセル (解除) させたいとき使用しています。

(14) Command

書式 : Command(コマンド名[引数])

“コマンド名”で指定したコマンドを実行するマクロです。また Command に記述できるコマンド名は Pete プログラムが用意しているコマンドの名前を指定し、必ず 1 つしか記述できません。Pete プログラムが用意しているコマンドについては次節 2 - 5 を参照してください。

(15) LoadCfgC

書式 : LoadCfgC(ソフトキーボード設定ファイル名)

デフォルトで Pete は Pete_key.cfg というファイル名のファイルがソフトキーボード設定ファイルであると定義しています。また、標準で提供しているソフトキーボード設定ファイル

名は「ひらがな.cfg」となっています。

独自のキーボードを作成する時は、これらのファイルの内容を変更してもできますが、標準キーボードと区別したい場合は、別ファイルに作成し、そのファイルをこのマクロで呼び出すようなソフトキーを標準キーボードに用意すれば、動的（Pete が動いている最中）にソフトキーボード設定ファイルを切り替えることができます。

ただし、色情報は引き継ぎませんので、新しい組み合わせの色を追加したいときは、色定義ファイル内（Pete_color.cfg）で追加してください。また、新規に作成したファイルが置かれているディレクトリを絶対または相対パスで指定しないと読み込めません。

ここで、キーボードが切り替わったとき、「読み」の内容と単語予測ウィンドウの内容がクリアされます。

(16) LoadCfgNc

書式：LoadCfgNc(ソフトキーボード設定ファイル名)

機能は(15)の“LoadCfgC”と同じですが、「読み」の内容と単語予測ウィンドウの内容がクリアされないこと（以前の状態が保持）が違います。

(17) SetFocus

書式：SetFocus (ソフトキーまたはグループの ID)

このマクロを記述したソフトキーが選択されたとき、スキャンが次にどこに行くかべきかを“ソフトキーまたはグループの ID”で指定します。SetFocus を記述していないソフトキーが選択されるとスキャンは必ず単語予測ウィンドウ・操作予測ウィンドウ・キーボードウィンドウのゾーン選択に行きます。

このゾーン選択に行かせたくないとき、標準で提供しているソフトキーボード設定ファイルの場合では、「ひらがな」キーボードの「拡張」キーなど画面が一時的に切り替わるとき、スキャンが切り替わった画面に行くように使用しています。つまり、「拡張」キーの OnSelected イベント発生時にこのマクロが実行されるようになっています。

(18) GetAllCandidates

書式：GetAllCandidates()

単語予測ウィンドウに表示する候補をフレーズ辞書、単語辞書、IME 辞書すべて検索対象としたものにするマクロです。原則ソフトキーから使うことはありません。

(19) GetImeCandidates

書式：GetImeCandidates()

単語予測ウィンドウに表示する候補を IME 辞書からのみ検索対象としたものにするマクロです。原則ソフトキーから使うことはありません。

2 - 5 . マクロ内コマンド記述 (Command)

(1) 概要

コマンドとはマクロの中のひとつの命令で、このコマンドの中には更に細かい命令が更に記述できます。コマンドとして用意されている詳細命令にはグループを隠したり、また表示させたりするなどソフトキーボードの動作を定義づけるものがあります。

コマンド名記述欄には必ず1つの引数しか記述できません。複数のコマンドを実行したいときは、マクロとしてコマンドを複数記述してください。コマンド1つ分の記述方式を以下に示します。

書式：コマンド名[引数]

但し、引数を必要としないコマンドのときは“ [引数] ”の部分は記述しません。また、引数はコマンドによって異なります。

コマンド名とその動作概要は以下の通りとなっています。

| コマンド | 動作概要 |
|---------------------|--------------------|
| ResetCounter | クリックカウンタをリセットする |
| RegisterWord | 単語登録ダイアログを出す |
| PeteQuit | 終了する |
| GoHome | フォーカスをホームに移す |
| GoKeyboard | フォーカスをキーボードに移す |
| GoYosoku | フォーカスを単語予測例示に移す |
| GoPYosoku | フォーカスを操作予測例示に移す |
| ToggleAutoScan | オートスキャンを On/Off する |
| NextCandidates | 単語予測の次の候補群を出す |
| PreviousCandidates | 単語予測の前の候補群を出す |
| PNextCandidates | 操作予測の次の候補群を出す |
| PPreviousCandidates | 操作予測の前の候補群を出す |
| Hide | キーを隠す |
| Show | キーを表示する |
| SmallKana | 小さい字を入力する |
| SetKeyboard | キーボードウィンドウを定義する |
| SetYosoku | 単語予測ウィンドウを定義する |
| SetPYosoku | 操作予測ウィンドウを定義する |
| SetHome | ホーム予ウィンドウを定義する |
| ClearRead | 読み文字列をクリアする |

(2) ResetCounter (引数なし)

書式：ResetCounter

P e t e は調査用に何回ソフトキーが入力されたかカウントしてファイルに記憶していますが、このカウンタをリセットするコマンドです。

(3) RegisterWord (引数なし)

書式：RegisterWord

単語登録のためのダイアログを開くコマンドです。

(4) PeteQuit (引数なし)

書式 : PeteQuit

P e t e プログラムを終了させるコマンドです。

(5) GoHome (引数なし)

書式 : GoHome

フォーカスをホームウィンドウとして定義されているウィンドウに移すコマンドです。

(6) GoKeyboard (引数なし)

書式 : GoKeyboard

フォーカスをキーボードウィンドウとして定義されているウィンドウに移すコマンドです。

(7) GoYosoku (引数なし)

書式 : GoYosoku

フォーカスを単語予測ウィンドウとして定義されているウィンドウに移すコマンドです。

(8) GoPYosoku (引数なし)

書式 : GoPYosoku

フォーカスを操作予測ウィンドウとして定義されているウィンドウに移すコマンドです。

(9) ToggleAutoScan (引数なし)

書式 : ToggleAutoScan

スキャンモードの開始 / 停止をするコマンドで、1 回実行するとスキャンモードが反転するコマンドです。

(10) NextCandidates

書式 : NextCandidates[候補番号]

次の候補群を単語予測ウィンドウに表示させるコマンドです。これは P e t e プログラム内で「単語予測ウィンドウ内に表示される“次候補”キー」の機能として使用され、原則ソフトキーボードからは使用しません。

(11) PreviousCandidates

書式 : PreviousCandidates[候補番号]

前の候補群を単語予測ウィンドウに表示させるコマンドです。これは P e t e プログラム内で「単語予測ウィンドウ内に表示される“前候補”キー」の機能として使用され、原則ソフトキーボードからは使用しません。

(12) PNextCandidates

書式 : PNextCandidates[候補番号]

次の候補群を操作予測ウィンドウに表示させるコマンドです。これは P e t e プログラム内で「操作予測ウィンドウ内に表示される“次候補”キー」の機能として使用され、原則ソフトキーボードからは使用しません。

(13) PPreviousCandidates

書式：PPreviousCandidates[候補番号]

前の候補群を操作予測ウィンドウに表示させるコマンドです。これはP e t eプログラム内で「操作予測ウィンドウ内に表示される“前候補”キー」の機能として使用され、原則ソフトキーボードからは使用しません。

(14) Hide

書式：Hide[ソフトキーまたはグループの ID]

指定したソフトキーまたはグループを隠す（非表示にすること）コマンドです。グループを指定した場合そのグループに所属するソフトキーとグループすべてが非表示になります。

(15) Show

書式：Show[ソフトキーまたはグループの ID]

指定したソフトキーまたはグループで非表示になっていたものを表示するコマンドです。グループを指定した場合そのグループに所属するソフトキーとグループすべてが表示されることとなります。

(16) SmallKana（引数なし）

書式：SmallKana

現在入力されている文字（1文字分）を小文字に変換します。但し、小文字のない文字は変換対象外となります。

(17) SetKeyboard

書式：SetKeyboard[ソフトキーまたはグループの ID]

どのグループがソフトキーボードウィンドウに相当するか動的に変更するコマンドです。また、ソフトキー ID も指定可能ですが通常指定しません。

(18) SetYosoku

書式：SetYosoku[ソフトキーまたはグループの ID]

どのグループが単語予測ウィンドウに相当するか動的に変更するコマンドです。また、ソフトキー ID も指定可能ですが通常指定しません。

(19) SetPYosoku

書式：SetPYosoku[ソフトキーまたはグループの ID]

どのグループが操作予測ウィンドウに相当するか動的に変更するコマンドです。また、ソフトキー ID も指定可能ですが通常指定しません。

(20) SetHome

書式：SetHome[ソフトキーまたはグループの ID]

どのグループがホームウィンドウに相当するか動的に変更するコマンドです。また、ソフトキー ID も指定可能ですが通常指定しません。

(21) ClearRead（引数なし）

書式：ClearRead

現在入力されている“読み”の文字（漢字変換の対象となっている入力文字）をすべてクリ

アするコマンドです。ただし、内部バッファのクリアのみで候補ウィンドウや読みの表示はクリアされません。

2 - 6 . その他

「2 - 1 . 書式」には当てはまらないがソフトキーボード設定ファイル内に記述でき、P e t e プログラム自身の動作の初期状態を設定するソフトキー設定用命令が存在します。このソフトキー設定用命令は全てのソフトキー、グループ定義の後（ファイルの最後）に記述してください。

(1) 書式

基本形：キーレイアウト用命令(ソフトキーまたはグループの ID)
以下にキーレイアウト用命令の一覧を示します。

| |
|-------------------|
| Home |
| Yomi |
| Yosoku |
| Pyosoku |
| Keyboard |
| Start |
| YosokuColor |
| PYosokuColor |
| NextCandidate |
| PreviosCandidate |
| Imecandidate |
| PNextCandidate |
| PPreviosCandidate |

(2) Home

書式：Home(ソフトキーまたはグループの ID)
P e t e 起動時どのグループがホームウィンドウに相当するかを決めるソフトキー設定用命令です。また、ソフトキーID も指定可能ですが通常指定しません。

(3) Yomi

書式：Yomi(ソフトキーまたはグループの ID)
P e t e 起動時どのグループが読み表示ウィンドウに相当するかを決めるソフトキー設定用命令です。また、ソフトキーID も指定可能ですが通常指定しません。

(4) Yosoku

書式：Yosoku(ソフトキーまたはグループの ID)
P e t e 起動時どのグループが単語予測ウィンドウに相当するかを決めるソフトキー設定用命令です。また、ソフトキーID も指定可能ですが通常指定しません。

- (5) PYosoku
書式 : PYosoku(ソフトキーまたはグループの ID)
P e t e 起動時どのグループが操作予測ウィンドウに相当するかを決めるソフトキー設定用命令です。また、ソフトキー ID も指定可能ですが通常指定しません。
- (6) Keyboard
書式 : Keyboard(ソフトキーまたはグループの ID)
P e t e 起動時どのグループがキーボードウィンドウに相当するかを決めるソフトキー設定用命令です。また、ソフトキー ID も指定可能ですが通常指定しません。
- (7) Start
書式 : Start(ソフトキーまたはグループの ID)
P e t e 起動時スキャンモードになっている (1 - 2 節 TimerFlag 変数による) 時、どのキーまたはグループからスキャンを開始するかを決めるキーレイアウト用命令です。また、通常はスキャンの開始 / 停止コマンドを記述したソフトキー ID を指定します。
- (8) YosokuColor
書式 : YosokuColor(色 ID)
単語予測ウィンドウの候補の色を色指定で定義した色パターン (ID) で指定します。
- (9) PYosokuColor
書式 : PYosokuColor(色 ID)
操作予測ウィンドウの候補の色を色指定で定義した色パターン (ID) で指定します。
- (10) NextCandidate
書式 : NextCandidate(色 ID)
単語予測ウィンドウ内の “ 次候補 ” キーの色を色指定で定義した色パターン (ID) を指定します。
- (11) PreviosCandidate
書式 : PreviosCandidate(色 ID)
単語予測ウィンドウ内の “ 前候補 ” キーの色を色指定で定義した色パターン (ID) を指定します。
- (12) Imecandidate
書式 : Imecandidate(色 ID)
予測ウィンドウ内の “ I M E 候補 ” キーの色を色指定で定義した色パターン (ID) を指定します。
- (13) PNextCandidate
書式 : PNextCandidate(色 ID)
操作予測ウィンドウ内の “ 次候補 ” キーの色を色指定で定義した色パターン (ID) を指定します。
- (14) PPreviosCandidate

書式：PPreviousCandidate(色 ID)

操作予測ウィンドウ内の“前候補”キーの色を色指定で定義した色パターン（ID）を指定します。

2 - 7 . 特別なグループ（ウィンドウとして扱われる）

グループを定義する時、Peteプログラムが予約しているグループ名を指定すると特別なウィンドウとして扱われます。以下に予約されているグループ名の一覧を示します。

| | |
|----------|------------|
| Home | ホームウィンドウ |
| Yosoku | 単語予測ウィンドウ |
| PYosoku | 操作予測ウィンドウ |
| Keyboard | キーボードウィンドウ |

（ 1 ）ホームウィンドウ

スキャンモードの時のスキャンホーム位置を示すウィンドウです。

（ 2 ）予測ウィンドウ

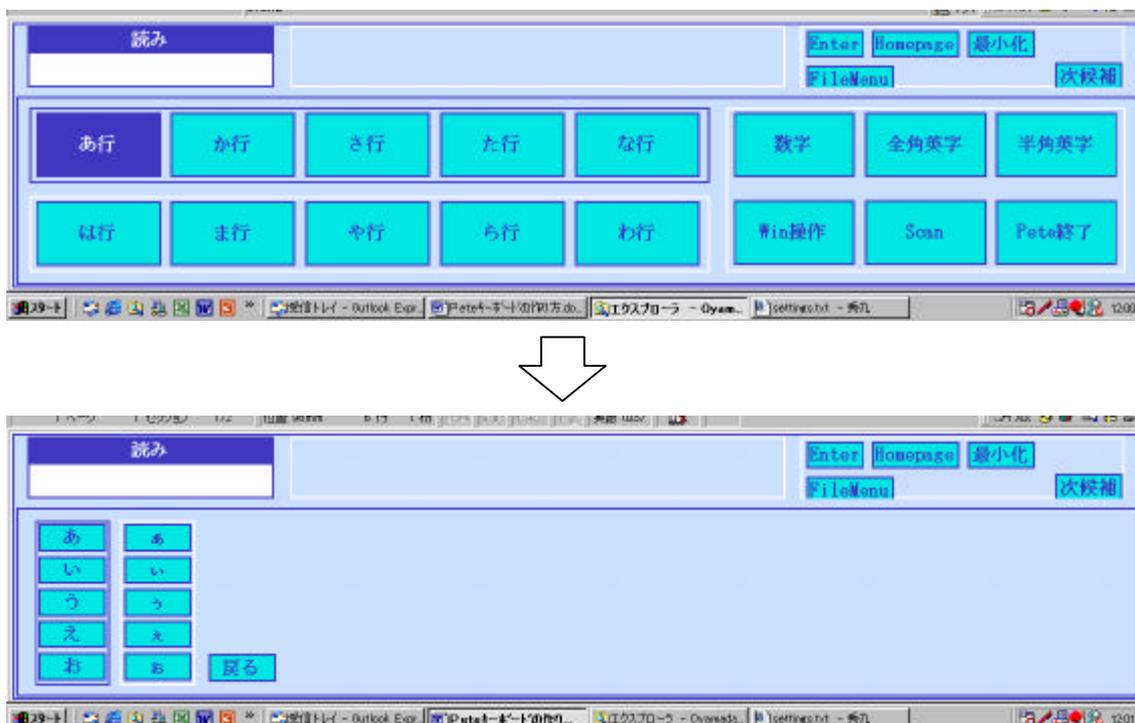
候補キーを表示する領域です。

（ 3 ）キーボードウィンドウ

ソフトキーボードを表示する領域です。

3 . サンプル初級者編

参考例 1) 簡単なキーボードを作る



3 - 1 . 動作環境の設定

「settings.txt」を開き、新たに作成するソフトキーボードを定義するファイル名を追加します。ここでは、ファイル名を「サンプル.cfg」とします。(加次イ マニュアル 1-2 参照)

KeyCfgFile=" サンプル.cfg"

この1行を書き加え、さらに他の KeyCfgFile の行をコメントアウト (先頭に「#」を記入) します。

また、ここでは絶対系座標 (加次イ マニュアル 1-3 (9)参照) を使用してキーレイアウトを行いますので、

MetricsMode = 2

となっていることを確認します。

3 - 2 . 色の設定

「pete_color.cfg」ファイルを開き、新たに設定するカラーセットを追加します。既存の Pete では Color(1) ~ Color(12)までを使用していますので、Color(101)からを使用してください。ここでは、新たに作成した以下の4つのカラーセットを使用します。(加次イ マニュアル 2-2 参照)

#グループの色

コメント行

Color(101){ Normal=cee1ff Focus=cee1ff Select=a5c9fe

背景色

```

fNormal=ffffff fFocus=413ac0 fSelect=413ac0 }           フレーム色
# 「読み」表示ラベルの色
Color(102){ Normal=413ac0 fNormal=413ac0 tNormal=ffffff }
# 「読み」表示ウィンドウの色
Color(103){ Normal=ffffff fNormal=413ac0 tNormal=000000 }
# キーの色
Color(104){ Normal=00eaea Focus=413ac0 Select=413ac0
fNormal=413ac0 fFocus=413ac0 fSelect=413ac0
tNormal=413ac0 tFocus=ffffff tSelect=ffffff }           文字色

```

3 - 3 . ソフトキーボード定義ファイルの作成

スキャンモードの際にフォーカスは、同じ「Parent」の中を、cfg ファイルに書かれている上から順に移動していきます。

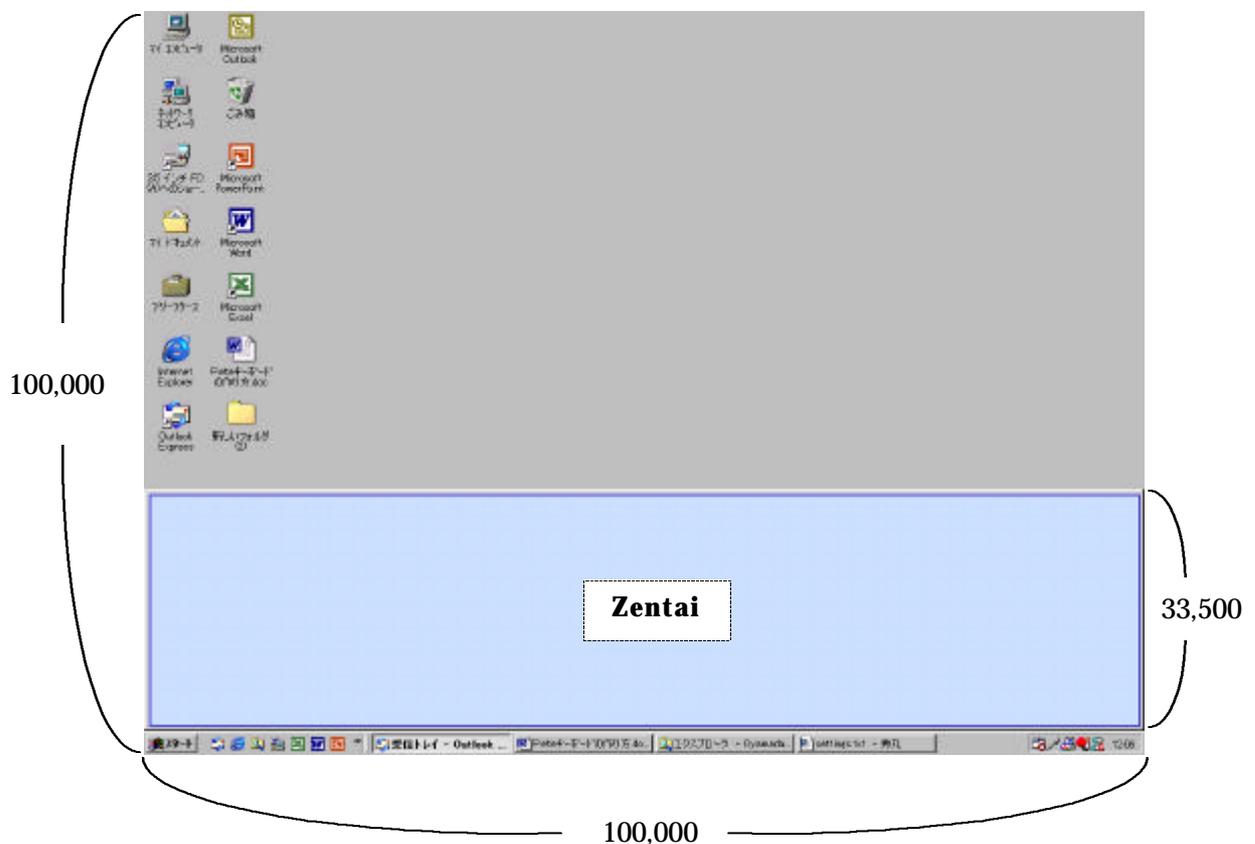
Pete 全体の画面の大きさを定義します。以下の Key(1)は規定のものとして記述します。
 (*加タイズ* マニュアル 2-3 参照)

#Pete 全体

```
Key(1){ Type=1 Parent=0 Name="Zentai" x=0 y=0 cx=100000 cy=33500 Colorset=101 Visible=1 }
```

横方向は「cx=99400」にするとフレーム4辺がきれいに表示されます。

図 3 - 1



読み表示ウィンドウ、単語予測ウィンドウ、操作予測ウィンドウ、キーボード領域を定義します。

ここでは、で定義した Zentai の左上を $x=0, y=0$ とします。(図 3 - 2)

Pete では、これから作ろうとする領域やキーの位置は、それらが属する Parent の左上の座標を常に「 $x=0, y=0$ 」、右下を「 $x=100000, y=100000$ 」とした相対座標で表します。

(図 3 - 3)

領域やキーの大きさはパソコンの画面全体を「 100000×100000 」とした絶対座標で表します。(図 3 - 4)

図 3 - 2



図 3 - 3

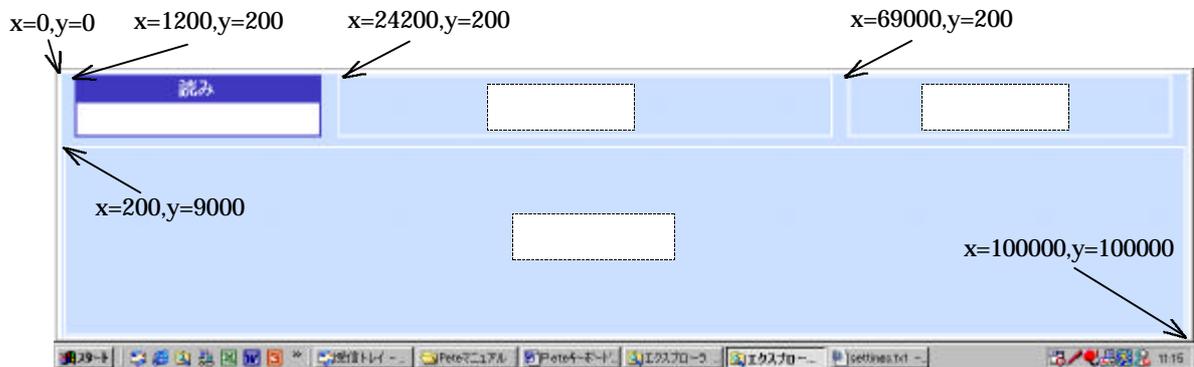
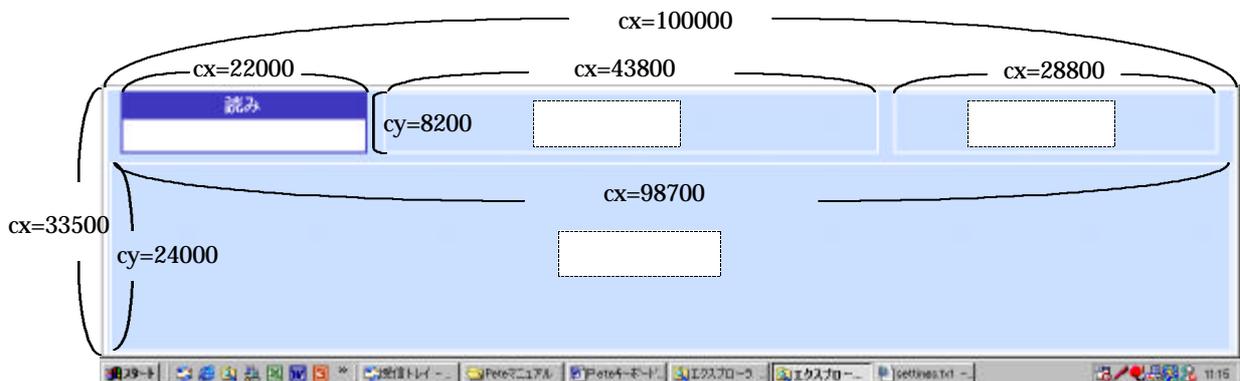


図 3 - 4



#読み表示ウィンドウ

**Key(4){ Type=6 Parent=1 Name="読み" x=1200 y=200 cx=22000 cy=4000
Colorset=102 Visible=1 }**

**Key(5){ Type=6 Parent=1 Name="" x=1200 y=3500 cx=22000 cy=5000
Colorset=103 Visible=1 }**

#単語予測ウィンドウ

**Key(6){ Type=1 Parent=1 Name="Yosoku" x=24200 y=200 cx=43800 cy=8200
Colorset=101 Visible=1 }**

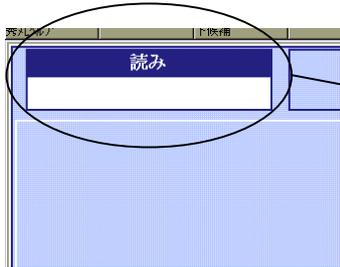
#操作予測ウィンドウ

**Key(7){ Type=1 Parent=1 Name="PYosoku" x=69000 y=200 cx=28800 cy=8200
Colorset=101 Visible=1 }**

#キーボード領域

**Key(8){ Type=1 Parent=1 Name="Keyboard" x=200 y=9000 cx=98700 cy=24000
Colorset=101 Visible=1 }**

図 3 - 5



Key(4)・・・「読み」ラベルを表示させるため
「Type=6 Name="読み"」とする。
Key(5)・・・実際に入力したカナを表示させるため
「Type=6 Name=""」とする。

キーボード領域中のグループを定義します。

ここでは、 で定義した Keyboard の左上を x=0、y=0 とします。

#ひらがなグループ

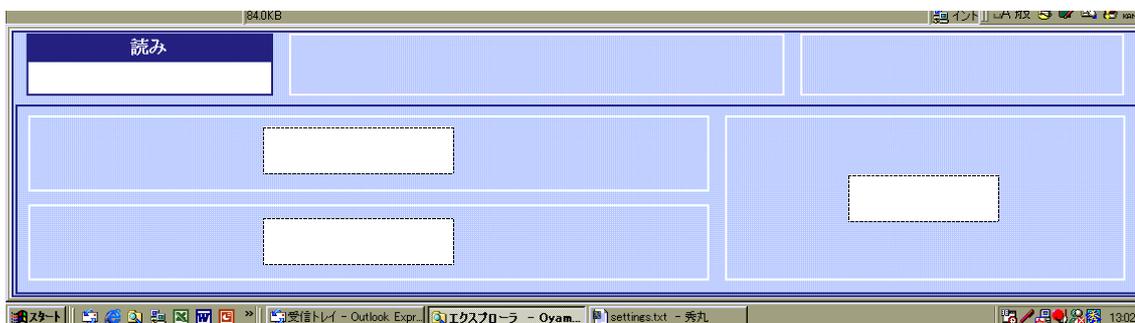
**Key(10){ Type=1 Parent=8 Name="Kanakey1" x=1000 y=1200 cx=60000 cy=10000
Colorset=101 Visible=1 }**

**Key(11){ Type=1 Parent=8 Name="Kanakey2" x=1000 y=12200 cx=60000 cy=10000
Colorset=101 Visible=1 }**

#P e t e 終了キーなどのグループ

**Key(12){ Type=1 Parent=8 Name="Sousa" x=62000 y=1200 cx=35500 cy=21000
Colorset=101 Visible=1 }**

図3 - 6



「あ行」「か行」等のキーを選択したときに表示されるひらがな入力キーボードの領域を定義します。位置と大きさは Key(8)で定義した Keyboard と同じに設定します。ただし、このひらがな入力キーボード領域は、一番初めの Pete 画面には現れないものなので、「Visible=0」とします。

これを「あ行」から「わ行」、「数字」、「全角英字」、「半角英字」の各キーについて定義します。(加タイズマニュアル 2-4、2-5 参照)

#ひらがな入力キーボード領域

#「あ行」キーを押したときの領域

```
Key(20){ Type=1 Parent=1 Name="Kanakey" x=200 y=9000 cx=98700 cy=24000
        Clorset=101
        onUnFocus="Command(Hide[20]) Command(Show[8])
        Command(SetKeyboard[8])" Visible=0 }
```

#「か行」キーを押したときの領域

```
Key(21){ Type=1 Parent=1 Name="Kanakey" x=200 y=9000 cx=98700 cy=24000
        Clorset=101
        onUnFocus="Command(Hide[21]) Command(Show[8])
        Command(SetKeyboard[8])" Visible=0 }
```

#「さ行」キーを押したときの領域

```
Key(22){ Type=1 Parent=1 Name="Kanakey" x=200 y=9000 cx=98700 cy=24000
        Clorset=101
        onUnFocus="Command(Hide[22]) Command(Show[8])
        Command(SetKeyboard[8])" Visible=0 }
```

⋮

フォーカスが外れた時の動作：現在表示されている領域（自分自身）を隠し、Key(8)の Keyboard を表示させ、Key(8)をキーボードに設定する。

「あ行」「か行」等のキーを定義します。 の Key(10)Kanakey1 の中に 5 つのキーを配置します。ここでは、Kanakey1 の左上を x=0、y=0 とします。

#ひらがな 1 行目

```
Key(101){ Type=3 Parent=10 Name="あ行" x=500 y=400 cx=11300 cy=8600
```

```
Colorset=104
```

```
onSelected="Command(Hide[8]) Command(Show[20])
```

```
Command(SetKeyboard[20]) SetFocus(20)" Visible=1 }
```

```
Key(102){ Type=3 Parent=10 Name="か行" x=12300 y=400 cx=11300 cy=8600
```

```
Colorset=104
```

```
onSelected="Command(Hide[8]) Command(Show[21])
```

```
Command(SetKeyboard[21]) SetFocus(21)" Visible=1 }
```

⋮

「あ行」キーを選択した時の動作：現在表示されている Key(8)の Keyboard を隠し、Key(20)の領域を表示させ、Key(20)をキーボードに設定し、且つスキャンはKey(20)の中で行う。

これを「あ行」から「な行」の各キーについて、また Key(11)Kanakey2 の中に配置する「は行」から「わ行」の各キーについても同様に定義します。

図 3 - 7



「数字」「Win 操作」等のキーボード切替キー、「Scan」「Pete 終了」キーを定義します。 の Key(12)Sousa の中に 6 つのキーを配置します。ここでは、Sousa の左上を x=0、y=0 とします。

#Pete 終了キーなど

```
Key(121){ Type=3 Parent=12 Name="数字" x=600 y=400 cx=11000 cy=8600
```

```
Colorset=104
```

```
onSelected="Command(Hide[8]) Command(Show[30])
```

```
Command(SetKeyboard[30])" Visible=1 }
```

```
Key(122){ Type=3 Parent=12 Name="全角英字" x=12100 y=400 cx=11000 cy=8600
```

```
Colorset=104
```

```
onSelected="Command(Hide[8]) Command(Show[31])
```

```
Command(SetKeyboard[31])" Visible=1 }
```

```
Key(123){ Type=3 Parent=12 Name="半角英字" x=23600 y=400 cx=11000 cy=8600
```

Colorset=104
onSelected="Command(Hide[8]) Command(Show[32])
Command(SetKeyboard[32])" Visible=1 }
Key(124){ Type=3 Parent=12 Name="Win 操作" x=600 y=11300 cx=11000 cy=8600
Colorset=104
onSelected="LoadCfgNc(Win 操作.cfg) " Visible=1 }
 既存の Win 操作キーボードを使用する。
Key(125){ Type=3 Parent=12 Name="Scan" x=12100 y=11300 cx=11000 cy=8600
Colorset=104 onSelected="Command(ToggleAutoScan)" Visible=1 }
 オートスキャンを On/Off する。
Key(126){ Type=3 Parent=12 Name="Pete 終了" x=23600 y=11300 cx=11000 cy=8600
Colorset=104 onSelected="Command(PeteQuit)" Visible=1 } Pete を終了する。

図 3 - 8



「あ行」キーを選択したときに表示されるひらがなキーのグループを定義します。「あ」「い」「う」「え」「お」のグループ（あ行 1）と、「ぁ」「ぃ」「ぅ」「ぇ」「ぉ」のグループ（あ行 2）の 2 つのグループを定義します。これは、 で定義した Key(20)の中に表示されます。ただし、Key(20)が表示されたのと同時にこのグループも表示されることになるので、「Visible=1」とします。

#あ行グループ

Key(201){ Type=1 Parent=20 Name="あ行 1" x=1500 y=1500 cx=7000 cy=21000
Colorset=101 Visible=1 }
Key(202){ Type=1 Parent=20 Name="あ行 2" x=9000 y=1500 cx=7000 cy=21000
Colorset=101 Visible=1 }

図 3 - 9



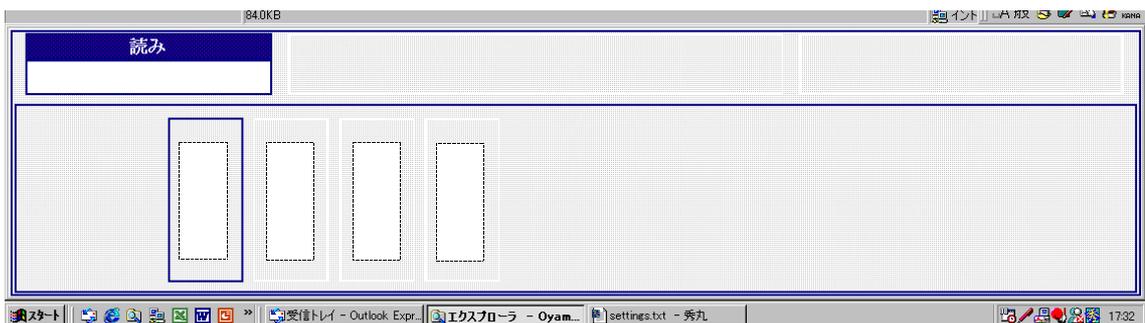
同様に、「か行」から「わ行」、「数字」、「全角英字」、「半角英字」についても定義します。「か行」のグループは、 で定義した Key(21)の中に表示されることになります。

#か行グループ

```
Key(211){ Type=1 Parent=21 Name="か行 1" x=13300 y=1500 cx=7000 cy=21000  
Colorset=101 Visible=1 }  
Key(212){ Type=1 Parent=21 Name="か行 2" x=20800 y=1500 cx=7000 cy=21000  
Colorset=101 Visible=1 }  
Key(213){ Type=1 Parent=21 Name="か行 3" x=28300 y=1500 cx=7000 cy=21000  
Colorset=101 Visible=1 }  
Key(214){ Type=1 Parent=21 Name="か行 4" x=35800 y=1500 cx=7000 cy=21000  
Colorset=101 Visible=1 }
```

⋮

図 3 - 10



「あ行」キーを選択したときに表示されるあ行の各キーを定義します。 で定義した Key(201)のあ行 1 グループの中に「あ」「い」「う」「え」「お」の 5 つのキーを配置します。ここでは、あ行 1 の左上を x=0、y=0 とします。ただし、Key(201)が表示されたのと同時にこのグループも表示されることになるので、「Visible=1」とします。

#あ行 1 のキー

```
Key(2011){ Type=3 Parent=201 Name="あ" x=200 y=200 cx=6300 cy=4000
```

Colorset=104 onSelected="AddRead(あ)" Visible=1 }

漢字変換対象に追加される。つまり読み表示ウィンドウ
に追加表示されるもの。

Key(2012){ Type=3 Parent=201 Name="い" x=200 y=4200 cx=6300 cy=4000

Colorset=104 onSelected="AddRead(い)" Visible=1 }

Key(2013){ Type=3 Parent=201 Name="う" x=200 y=8200 cx=6300 cy=4000

Colorset=104 onSelected="AddRead(う)" Visible=1 }

Key(2014){ Type=3 Parent=201 Name="え" x=200 y=12200 cx=6300 cy=4000

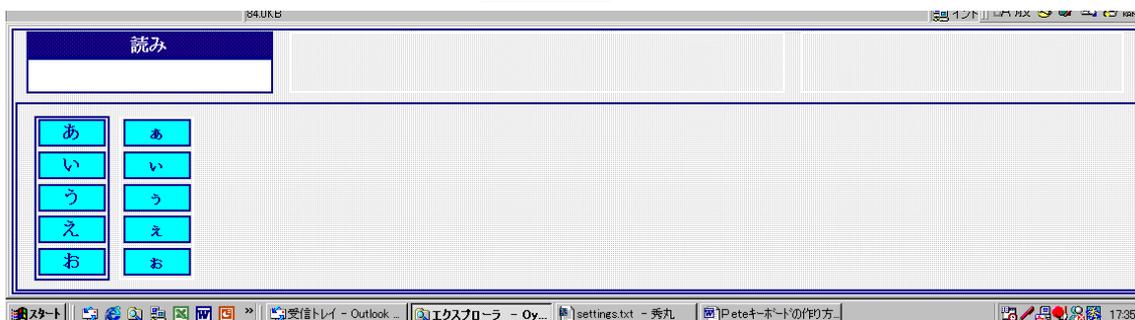
Colorset=104 onSelected="AddRead(え)" Visible=1 }

Key(2015){ Type=3 Parent=201 Name="お" x=200 y=16200 cx=6300 cy=4000

Colorset=104 onSelected="AddRead(お)" Visible=1 }

同様に「あ」から「お」についても定義します。

図 3 - 1 1



で定義したあ行キーの画面から元のキーボードに戻るために「戻る」キーを定義します。

で定義した Key(20)の中で、 の 2 つのグループの右隣に配置します。

#あ行の「戻る」キー

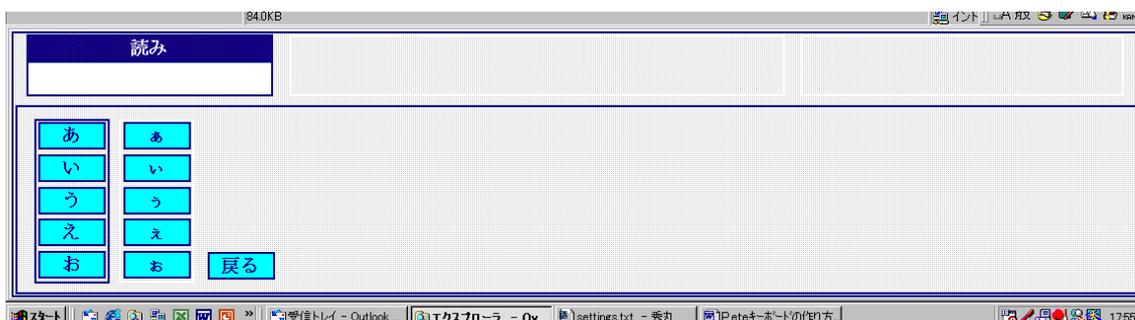
Key(2035){ Type=3 Parent=20 Name="戻る" x=16700 y=18000 cx=6300 cy=4000

Colorset=104

onSelected="Command(Hide[20]) Command(Show[8])

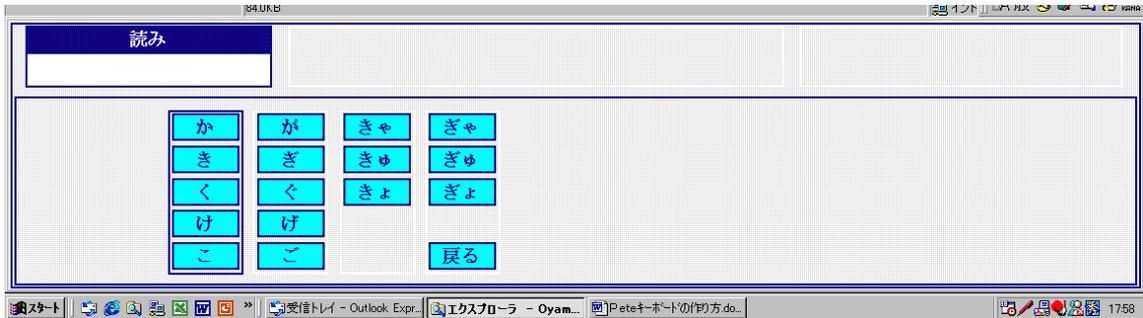
Command(SetKeyboard[8])" Visible=1 }

図 3 - 1 2



「か行」から「わ行」についても、 を繰り返してそれぞれのキーを定義します。

図3-13



「数字」「全角英字」「半角英字」についても、 を繰り返してそれぞれのキーを定義します。ただし、これらのキーは直接アプリケーションに入力される文字であるため、以下のように定義します。

#数字入力キー

**Key(3011){ Type=3 Parent=30 Name="1" x=200 y=300 cx=6600 cy=5200
Colorset=4 onSelected="DirectPhrase(1)" Visible=1 }**

アプリケーションに直接入力される。

**Key(3012){ Type=3 Parent=30 Name="2" x=200 y=5500 cx=6600 cy=5200
Colorset=4 onSelected="DirectPhrase(2)" Visible=1 }**

#全角英字キー

**Key(3111){ Type=3 Parent=31 Name="A" x=200 y=300 cx=6600 cy=4450
Colorset=4 onSelected="DirectPhrase(A)" Visible=1 }**

**Key(3112){ Type=3 Parent=31 Name="A" x=200 y=4750 cx=6600 cy=4450
Colorset=4 onSelected="DirectPhrase(A)" Visible=1 }**

#半角英字キー

**Key(3211){ Type=3 Parent=32 Name="a" x=200 y=300 cx=6600 cy=4450
Colorset=4 onSelected="DirectPhrase(a)" Visible=1 }**

**Key(3212){ Type=3 Parent=32 Name="b" x=200 y=4750 cx=6600 cy=4450
Colorset=4 onSelected="DirectPhrase(b)" Visible=1 }**

⋮
⋮

最後に、以下の10行を宣言します。(加算バリエーション 2-6 参照)

- | | | | |
|-------------------|---|--------------------|------------------------------------|
| スキャンモードに 必要な宣言 | { | Home(6) | ・・・スキャンモードの時、キーを選択した後にスキャンが移るところ |
| | | Keyboard(8) | ・・・Pete 立ち上げ時に表示されるキーボード領域 |
| | | Start(125) | ・・・Pete を立ち上げた時、スキャンが始まるキー (図3-14) |
| | | Yomi(5) | ・・・入力した文字を表示させるところ |

- 色の宣言
- Yosoku(6)** . . . 単語予測を表示させるところ
 - PYosoku(7)** . . . 操作予測を表示させるところ
 - YosokuColor(104) . . . 単語予測、操作予測の候補キーのカラーセット
 - Nextcandidate(104)** . . . 「次候補」キーのカラーセット
 - Previouscandidate(104)** . . . 「前候補」キーのカラーセット
 - imecandidate(104)** . . . 「IME 候補」キーのカラーセット

図 3 - 14



4. サンプル上級者編

参考例) 自作キーボードと標準キーボードを合わせて使用する。

「新ひらがなキーボード」



「英字キーボード」(Pete 標準キーボード)



「Excel 専用キーボード」



「Win 操作キーボード」(Pete 標準キーボ



4 - 1 . キーを追加する

3 . サンプル初級者編で作成したキーボードに以下のキーを追加します。

「BS」・・・読みウィンドウに文字があれば最後に入力した1文字から削除し、続けて、
又は読みウィンドウに文字がなければアプリケーションの文字カーソル手前
の1字を削除するキーです。

「濁点」「半濁点」・・・直前に入力した読みウィンドウ内の文字に濁点や半濁点をつけ
るキーです。

「小字」・・・直前に入力した読みウィンドウ内の文字を小文字に変換します。

「単語登録」・・・単語登録のダイアログを開きます。

「Excel 起動」・・・Excel を起動し、同時に Excel 専用のキーボードに切り替えます。

「Excel キーボード」・・・キーボードだけを Excel 専用のものに切り替えます。

「Excel 終了」・・・Excel を終了させます。

図 4 - 1 3 . サンプル初級者編で作成したキーボード



「BS」キー

```
Key(121){ Type=3 Parent=12 Name="BS" x=600 y=400 cx=10300 cy=4700  
Colorset=104  
onSelected="Direct(BS)" Visible=1 }
```

「濁点」「半濁点」キー

```
Key(122){ Type=3 Parent=12 Name="濁点" x=600 y=5300 cx=10300 cy=4700  
Colorset=104  
onSelected="AddRead(^)" Visible=1 }
```

```
Key(123){ Type=3 Parent=12 Name="半濁点" x=600 y=10200 cx=10300 cy=4700  
Colorset=104  
onSelected="AddRead(^)" Visible=1 }
```

「小字」キー

```
Key(124){ Type=3 Parent=12 Name="小字" x=600 y=15100 cx=10300 cy=4700  
Colorset=104  
onSelected="Command(SmallKana)" Visible=1 }
```

「単語登録」キー

```
Key(133){ Type=3 Parent=13 Name="単語登録" x=12100 y=400 cx=11000 cy=8600
```

Colorset=104
onSelected="Command(RegisterWord)" Visible=1 }

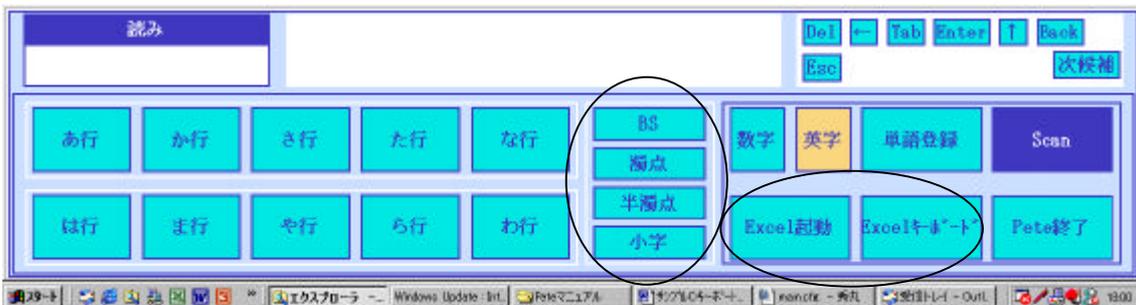
「Excel 起動」キー

Key(135){ Type=3 Parent=13 Name="Excel 起動" x=600 y=11300 cx=11000 cy=8600
Colorset=104 onSelected="LoadCfgC(excel.cfg) Command(SetYosoku[7])
Command(SetPYosoku[7]) Command(GoHome) Process(C:\¥¥Program
Files¥¥Microsoft Office¥¥Office¥¥Excel¥.exe)" Visible=1 }

「Excel キーボード」キー

Key(136){ Type=3 Parent=13 Name="Excel キーボード" x=12100 y=11300 cx=11000
cy=8600
Colorset=104 onSelected="LoadCfgC(excel.cfg) Command(SetYosoku[7])
Command(SetPYosoku[7]) Command(GoHome) " Visible=1 }

図4 - 2 キーを追加したキーボード



「Excel 終了」キー

Key(252){ Type=3 Parent=8 Name="Excel 終了" x=14400 y=200 cx=14000 cy=4400
Colorset=104 onSelected="Kill(Microsoft Excel) Wait(20)
LoadcfgNc(main.cfg) Command(SetYosoku[6]) Command(SetPYosoku[7])
Command(GoHome)" Visible=1 }

図4 - 3 「Excel 終了」キーを使用したキーボード



4 - 2 . 単語予測ウィンドウを使用しない

図4 - 4 単語予測ウィンドウを使用しないキーボード



図4 - 4のように単語予測ウィンドウを使用しないキーボードも作ることが出来ます。その場合は、単語予測ウィンドウの定義や、cfg ファイルの最後で Yosoku()の宣言をする必要もありません。

4 - 3 . 標準キーボードと相互に切り替えて使用するキーを追加する。

Pete の標準キーボードと自作の2つのキーボードを相互に切り替えながら使用できるようにします。ここでは、標準キーボードの

- ・ ひらがな.cfg
- ・ Win 操作.cfg
- ・ 全角大文字.cfg
- ・ 半角大文字.cfg
- ・

と自作のキーボード

- ・ main.cfg・・・新ひらがなキーボード
- ・ excel.cfg・・・Excel 専用キーボード

の6つのcfgファイルを使用します。

自作の「新ひらがなキーボード」で「英字」キーを選択すると、標準キーボードの「全角大文字キーボード」に切り替わるようになります。

図4 - 5 標準「全角大文字キーボード」に切り替えるキー



<main.cfg>

```
Key(132){ Type=3 Parent=13 Name="英字" x=6350 y=400 cx=5250 cy=8600
Colorset=106
onSelected="LoadCfgNc(全角大文字.cfg) Command(SetYosoku[6])
Command(SetPYosoku[7]) Command(GoHome)" Visible=1 }
```

自作の「Excel 専用キーボード」で「英字」キーを選択すると、標準キーボードの「半角大文字キーボード」に、「Win 操作」キーを選択すると標準キーボードの「Win 操作キーボード」に切り替わるようにします。

図4 - 6 標準「半角大文字キーボード」「Win 操作キーボード」に切り替えるキー



<excel.cfg>

```
Key(254){ Type=3 Parent=8 Name="英字" x=200 y=4500 cx=14000 cy=4400
Colorset=106 onSelected="LoadCfgNc(半角大文字.cfg)
Command(SetYosoku[4]) Command(SetPYosoku[5])
Command(GoHome)" Visible=1 }
Key(256){ Type=3 Parent=8 Name="Win 操作" x=28600 y=4500 cx=14000 cy=4400
Colorset=106 onSelected="LoadCfgNc(Win 操作.cfg)
Command(SetYosoku[4]) Command(SetPYosoku[5])
Command(GoHome)" Visible=1 }
```

「全角大文字キーボード」から自作の「新ひらがなキーボード」「Excel 専用キーボード」に切り替わるように cfg ファイルを書き換えます。

図4 - 7 「変更前」全角大文字キーボード



<全角大文字.cfg>

**Key(194){ Type=3 Parent=24 Name=" a b c " x=200 y=11250 cx=9100 cy=3750
Colorset=8 Visible=1
onSelected="LoadCfgNc(全角小文字.cfg)" }**

**Key(195){ Type=3 Parent=24 Name="abc" x=9300 y=11250 cx=9100 cy=3750
Colorset=8 Visible=1
onSelected="LoadCfgNc(半角小文字.cfg)" }**



図4 - 8 「変更後」全角大文字キーボード



<全角大文字.cfg>

**Key(194){ Type=3 Parent=24 Name="新ひらがな" x=200 y=11250 cx=9100 cy=3750
Colorset=107 Visible=1
onSelected="LoadCfgNc(main.cfg) Command(SetYosoku[6])
Command(SetPYosoku[7]) Command(GoHome)" }**

**Key(195){ Type=3 Parent=24 Name="Excelキ" x=9300 y=11250 cx=9100 cy=3750
Colorset=107 Visible=1
onSelected="LoadCfgNc(excel.cfg) Command(SetYosoku[7])
Command(SetPYosoku[7]) Command(GoHome)" }**

標準キーボードの他の3つのcfgファイルについても、同様に書き換えます。

ただし、「操作予測ウィンドウ」内の「候補キー」の色はうまく切り替わらないことがあります。