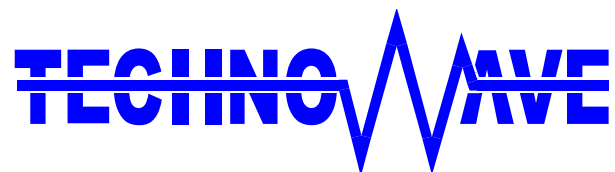


簡タッチ ユーザーズマニュアル

Ver1.5対応



テクノウェーブ株式会社

目次

1. はじめに	3
1.1. 注意事項.....	3
1.2. 商標.....	3
1.3. 簡タッチ EXE のユーザー様へのご注意.....	3
2. 製品概要	4
2.1. 特徴.....	4
2.2. 動作環境.....	4
3. 以前のバージョンとの相違	5
4. インストール方法	6
4.1. ファイル構成.....	6
4.2. インストール作業.....	6
5. 起動と終了	7
6. 動作説明	8
6.1. キー入力.....	8
6.2. メイン画面.....	9
6.3. リスト表示画面.....	12
6.4. 動作モード.....	13
7. カスタマイズ	14
7.1. ウィンドウ枠の状態.....	16
7.2. タイトルバーの状態.....	19
7.3. リスト表示キーの表示.....	21
7.4. 入力文字数の制限.....	22
7.5. 外部／内部モードの切り替え.....	23
7.6. ウィンドウ移動の可否.....	24
7.7. IME ON/OFF キーの表示.....	25
7.8. ウィンドウの位置と大きさ.....	26
7.9. ウィンドウのキャプション.....	28
7.10. ウィンドウのアイコン.....	29
7.11. 文字表示枠への文字列設定.....	30
7.12. リスト設定.....	31
7.13. キャプション.....	32

7.14. フォント.....	35
7.15. 色.....	37
7.16. 文字コード.....	39
7.17. オブジェクト番号について.....	42
7.18. デフォルトの設定について.....	44
8. イベント通知機能.....	46
8.1. 通知方式.....	46
8.2. 通知理由.....	47
8.3. アプリケーションからの戻り値による動作.....	49
9. 状態の記録と復帰.....	50
9.1. 記録範囲.....	50
9.2. 記録先について.....	51
10. 関数.....	53
10.1. 関数一覧.....	53
10.2. 変数型定義.....	54
10.3. 関数詳細.....	59
11. APPENDIX.....	87
11.1. コード例.....	87
12. 既知の不具合.....	89
12.1. IME 制御不具合.....	89
12.2. .NET 版 VISUAL BASIC で内部モードの不具合.....	89
13. サポート情報.....	90

1. はじめに

このたびは『簡タッチ』をご利用頂き、まことにありがとうございます。注意事項をよくお読みになり、ご使用いただけますようお願い申し上げます。

1.1. 注意事項

「12 既知の不具合」において、簡タッチの不具合状況を記載しています。ご利用の前に必ずご確認ください。

- 導入の前に本ソフトウェアがご利用の条件において正しく動作するか必ず確認して下さい。
- 本書の内容の一部または全部を無断転載することをお断りします。
- 本書の内容については、将来予告なしに変更することがあります。
- 本書の内容につきましては万全を期して作成しておりますが、万一誤りまたはお気づきの点がございましたら、弊社にご連絡くださいますようお願い致します。
- 本ソフトウェアおよび本書を運用した結果について、上記にかかわらず一切の責任を負いかねますのでご了承ください。

1.2. 商標

「Windows」「Visual C++」は米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

「ATOK」は株式会社ジャストシステムの登録商標(商標)です。

1.3. 簡タッチ EXE のユーザー様へのご注意

簡タッチ EXE (ファイル名が KtouchFE.exe) の参照資料としてお使いになるユーザー様は、以下にご注意下さい。

- ・カスタマイズ用の資料として参照頂けるのは、以下の項目となります。
 - 6 動作説明
 - 7 カスタマイズ
 - 9 状態の記録と復帰
- ・簡タッチ EXE は外部モードで使用するものなので、内部モードに関する内容については参照できません。
- ・簡タッチ EXE (KtouchFE.exe) に付属している簡タッチ (KtouchFE.dll) は、専用の DLL であり、KtouchFE.exe と切り離して単独で利用する事はできません。

2. 製品概要

2.1. 特徴

『簡タッチ』は、タッチパネルでの文字入力操作を想定した、Windows®上で動作するソフトキーボードです。本製品の本体はダイナミックリンクライブラリ(DLL)形式ですので、アプリケーションに組み込んでご使用いただけます。主な特徴は以下のとおりです。

■ タッチパネルでの操作を意識した画面

- ・押しやすいキーの大きさ : 出来る限りキーを大きくしました。
- ・分かりやすいキー配置 : 50音順、ABC順です。

■ 組み込み利用を考えた機能

- ・カスタマイズ機能 : ウィンドウ枠・色・フォント・文字コード等を変更可能です。
- ・文字リスト選択機能 : 文字リストから選んで入力できます。
- ・イベント通知機能 : 簡タッチ上の様々なイベントがアプリケーション側に通知され、動作を制御することができます。

2.2. 動作環境

本製品の動作環境を以下に示します。

表 1 動作環境

対応 OS	日本語版 Windows 2000,XP,Vista
モジュール形態	ダイナミックリンクライブラリ(DLL)
ファイル名	Ktouch.dll
対応日本語システム	MS-IME (ATOK®の場合は、文字コードのカスタマイズ必要になる場合があります)

3. 以前のバージョンとの相違

以前のバージョン (Ver1.4) に対して以下の相違があります。詳細については本書の該当する箇所をご覧ください。

■ IME ON/OFF キーと変換キーを追加

IME ON/OFF キーと変換キーを新たに設けました。アクション `KTCA_IMEONOFF` と `KTCA_CONVBTN` で、キーの表示／非表示を切り替えることが可能です。また、このアクションは、状態の記録と復帰にも対応しています。(記録範囲 `KTCSV_ACTION` に含まれます)

■ テーブル番号を直接設定する方法を追加

アプリからテーブル番号を設定する関数 `KTC_SetTableNum` を追加しました。また、現在のテーブル番号を読み出す `KTC_GetTableNum` も同時に追加しました。

■ テーブル番号を記録・復帰する方法を追加

テーブル番号を記録・復帰するため、新たに記録範囲 `KTCSV_TABLE` を追加しました。これにより、カスタマイズ等でもスタート時のテーブル指定ができるようになりました。

4. インストール方法

4.1. ファイル構成

本製品は、以下のファイルにて構成されます。

表 2 ファイル構成

簡タッチ本体	Ktouch.dll
各言語用ヘッダーファイル	Ktouch.h : C、C++用 Ktouch.bas : VB6.0 用 Ktouch.vb : VB.NET 用
リンク用 lib ファイル	Ktouch.lib

4.2. インストール作業

簡タッチをインストールするための特別なインストーラは用意していません。

「4.1 ファイル構成」で示されるファイルを、組み込むアプリケーションのプロジェクトに組み込んでお
使い下さい。

5. 起動と終了

アプリケーションから簡タッチを起動する手順を以下に示します。

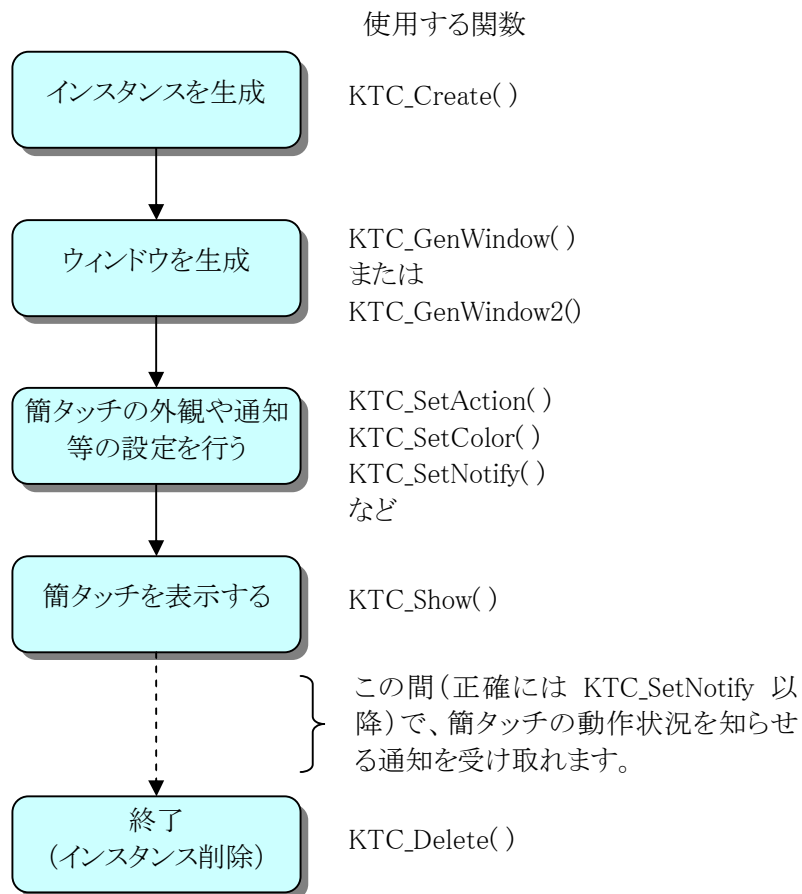


図 1 起動から終了までの概略フロー

◇ 起動

起動は、KTC_Create 関数、KTC_GenWindow 関数を順次実行してください。その後、簡タッチの外観やイベント通知等の必要な設定を行い、KTC_Show 関数で表示します。

ただし、外観等の変更は簡タッチ表示中においても実行可能です。

◇ 終了

終了する場合、必ず KTC_Delete 関数にてインスタンス削除を行わなければなりません。

※ 簡タッチ上のクローズボタン(タイトルバー上の×のボタン)をクリックしても画面が隠れるだけで、インスタンスが削除された訳ではありません。終了時は必ず KTC_Delete 関数を実行してください。

6. 動作説明

6.1. キー入力

KTC_Create 関数でインスタンスを生成した際、簡タッチは起動のためのキー入力を要求するため、下図のキー入力画面を表示します。

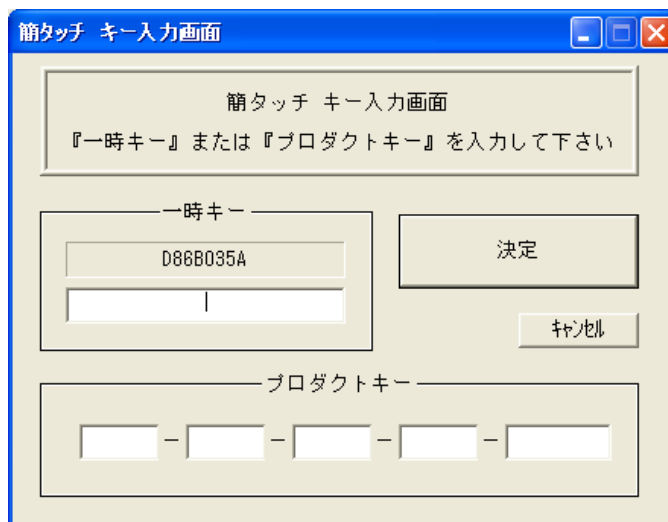


図 2 キー入力画面

ここで、一時キーの枠内に表示されている英数字8文字(ここでは 56486E1A)を、直ぐ下のボックスにキー入力して「決定」をクリックすれば、簡タッチが起動します。

一時キーは簡タッチが終了されるまで有効なので、起動する毎にキー入力を求めてきます。

ライセンスを購入された場合は、ライセンスで受けたキーをプロダクトキーの枠内に入力します。

(キー入力は管理者権限のアカウントで行ってください)

プロダクトキーを一度入力すると、次回からはキー入力画面は表示されず、直接簡タッチが起動します。

6.2. メイン画面

簡タッチを立ち上げた直後に表示されるメイン画面(内部モード時)を下図に示します。(外部モードには「文字表示枠」「リスト表示キー」「入力通知キー」がありません。詳しくは、「6.4 動作モード」をご覧ください)

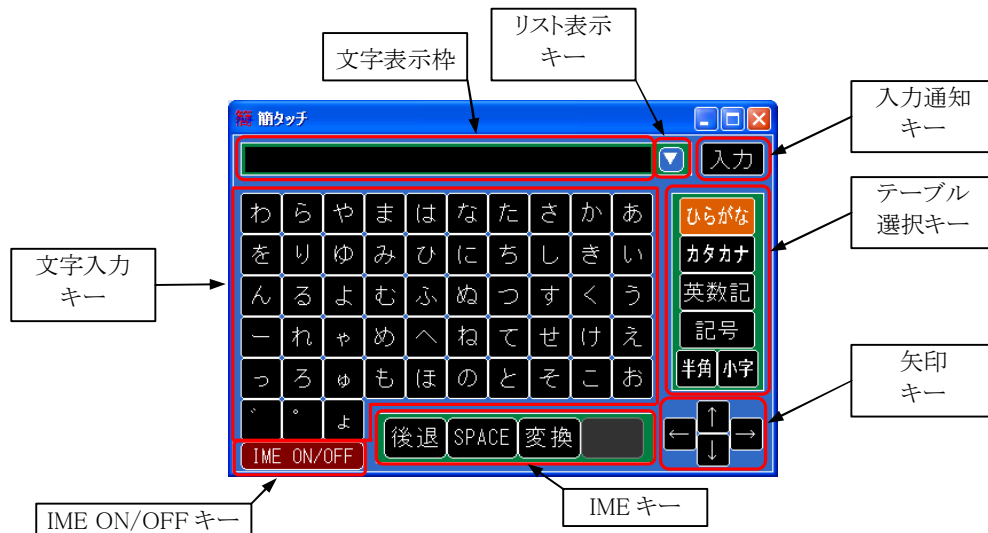


図 3 内部モードのメイン画面

◇ テーブル選択キー



文字入力キーに割り当てる文字テーブルを選択するキーです。「ひらがな」、「カタカナ」、「英数記」、「記号」の4つのキーはメインテーブル選択キーと呼ばれ、どれか1つを選択できます。「半角」「小字」の2つのキーはサブテーブル選択キーと呼ばれ、独立して選択ができ、メインテーブル選択キーと組み合わせて使用します。

左図で選択されているキーは、オレンジ色になっています。

右図は、半角カタカナに切り替えた例です。



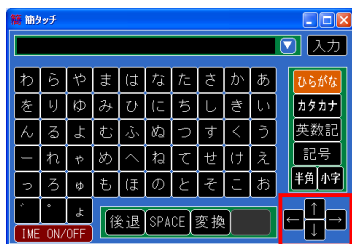
図 4 テーブル選択キーによる表示例

◇ 文字入力キー



合計53個のキーは、文字入力をするためのキーです。
これらのキーは、テーブル選択キーにより、カタカナや記号等に切り替えて入力することが出来ます。

◇ 矢印キー



この4つのキーは、ハードウェアキーボードの矢印キーと同等の動作を行いますので、カーソルの移動や変換中の候補選択等に使用します。

◇ IMEキー



この3つのキーは、ハードウェアキーボードの「後退(バックスペース)」「スペース」「変換」「Enter」と同等の動作を行います。
ただし「Enter」キーは外部モードの時のみ表示されます。(下図参照)



図 5 非変換中の IME キー

「スペースキー」は IME の初期設定により、変換文字入力中は「変換キー」として、変換文字入力中でない場合は「スペース文字キー」として機能します。また、スペース文字キーの入力は、テーブル選択キーの「半角」の ON/OFF により、半角スペースまたは全角スペースとなります。

注意: スペースキーに対する IME の初期設定を変更すると正しく機能しない場合があります

以前のバージョンのように、「スペースキー」と「Enter」キーの文字を無変換中と変換中で変えることもできます。V1.5以降では、同じ文字が初期値となるように変更されました。これは IME 制御が正しく行なえないアプリに対して、表示が正しく切り替わらない場合があるためです。

以前のバージョンとの互換性のため、「変換キー」は非表示とすることができます。(下図)



図 6 変換キーの表示切替

表示切り替えについては、「7 カスタマイズ」を参照して下さい。

◇ IME ON/OFF キー



このキーは、手動でIME を ON/OFF するためのものです。簡タッチは通常、文字入力キーをクリックした場合に自動的にIME の制御を行います。IME の制御が効かないアプリケーションの場合に手動でIME を ON/OFF する事ができます。

以前のバージョンとの互換性のため、「IME ON/OFF キー」は非表示とすることができます。表示切り替えについては、「7 カスタマイズ」を参照して下さい。

◇ 文字表示枠



文字入力キーによって入力された文字列は、この文字表示枠に表示されます。(内部モードの時)

ここに表示された文字列を取得するには、簡タッチの関数を使用します。

◇ リスト表示キー



このキーをクリックすると、文字列の一覧がリスト表示されます。(「6.3 リスト表示画面」をご覧ください)

ユーザーはリストから文字列を選択することができるので、入力作業を減らすことができます。

◇ 入力通知キー



このキーをクリックすると、アプリケーションへ入力通知キーがクリックされた事をイベント通知します。そのため、アプリケーションが文字表示枠の文字列を取得するタイミングとして使用できます。

イベント通知の詳細については「8 イベント通知機能」をご覧ください。

6.3. リスト表示画面

リスト表示キーをクリックすると、以下に示すようにリスト表示が行われます。ユーザーは、リストの中から任意の一行を選択し、「決定」キーをクリックすることにより、選択した文字を自動的に入力できます。入力した文字は、文字表示枠の現在のカーソル位置に挿入されます。リスト表示を終了するには、「決定」キーもしくは「キャンセル」キーをクリックします。



図 7 リスト表示画面

◇ リストの設定方法

リストの設定はアプリケーションが行います。リストを設定するタイミングは、リストを表示する直前までに簡タッチの関数を使用して行います。リスト表示ボタンがクリックされると、リスト表示前にアプリケーションにイベント通知されるので、そのハンドラ中でリスト設定を行うことも可能です。

リストの内容は、簡タッチを終了するか、再度リスト設定するまで変わりません。

6.4. 動作モード

『簡タッチ』は大きく分けて、キー入力した文字の送り先によって2つのモードを有しています。以下は、動作モードによる画面の違いを示しています。



図 8 動作モードによる画面の違い

※ 動作モードを変更するには、KTC_SetAction 関数で行います。

◇ 外部モード

入力された文字は現在のアクティブウィンドウに送られます。アクティブウィンドウとは、キーボード入力を受け付けているウィンドウの事です。たとえば、Notepad を起動して、その画面をマウスなどで選択すると、以降の簡タッチのキー入力による文字は、この Notepad に書き込まれます。

◇ 内部モード

入力された文字は簡タッチ上の文字表示枠に書き込まれます。「入力通知キー」をクリックするとアプリケーションにクリックされた事を知らせるイベント通知がされるので、文字列の取得タイミングとして使用することができます。文字表示枠に書き込まれた文字列を取得するには、簡タッチの KTC_GetText 関数を使用します。

■ 外部モード2について

Ver1.4 からは外部モード2の設定が出来るようになりました。従来の外部モードの上に「閉じるキー」が表示されます。

「閉じるキー」は内部モードの「入力通知キー」と全く同様の動作をします。したがって「閉じるキー」をクリックするとイベント KTCR_BTNINPUT が発生します。



7. カスタマイズ

簡タッチは、キーの色、フォント、キー入力時の送信コードなど、様々なカスタマイズが可能です。このカスタマイズ機能は関数としてエクスポートされており、アプリケーション側から呼び出すことにより実現できます。下表は、カスタマイズ項目と利用する設定関数を示しています。

表 3 カスタマイズ項目と設定関数

No.	カスタマイズ項目	利用する設定関数	備考
1	ウィンドウ枠の状態	KTC_SetAction	KTCA_FRAME 使用。
2	タイトルバーの状態		KTCA_TITLEBAR 使用
3	リスト表示キーの表示		KTCA_LISTBTN 使用
4	入力文字数の制限		KTCA_LIMITTEXT 使用
5	外部／内部モードの切り替え		KTCA_EXTERNAL 使用
6	ウィンドウ移動の可否		KTCA_MOVEWINDOW 使用
7	IME ON/OFF キーの表示		KTCA_IMEONOFF 使用
8	変換キーの表示		KTCA_CONVBTN 使用
9	ウィンドウの位置と大きさ	KTC_SetDlgPosition	
10	ウィンドウのキャプション	KTC_SetDlgCaption	
11	ウィンドウのアイコン	KTC_SetIcon	
12	文字表示枠への文字列設定	KTC_SetText KTC_SetText2	
13	リスト設定	KTC_SetList	
14	キャプション	KTC_SetCaption	これらの関数は実行後に KTC_UpdateViewによる画面更新 が必要です。
15	フォント	KTC_SetFont	
16	色	KTC_SetColor	
17	文字コード	KTC_SetKeyCode	
18	テーブル番号	KTC_SetTableNum	

また、カスタマイズ状態を読み出すため、各設定関数に対する読出し関数が用意されています。

表 4 カスタマイズ状態を読み出す関数

No.	設定関数	対応する読出し関数
1	KTC_SetAction	KTC_GetAction
2	KTC_SetDlgPosition	KTC_GetDlgPosition
3	KTC_SetText,KTC_SetText2	KTC_GetText
4	KTC_SetCaption	KTC_GetCaption
5	KTC_SetFont	KTC_GetFont
6	KTC_SetColor	KTC_GetColor
7	KTC_SetKeyCode	KTC_GetKeyCode
8	KTC_SetTableNum	KTC_GetTableNum

※KTC_SetDlgCaption,KTC_SetIcon,KTC_SetList に対応する読出し関数はありません。

7.1. ウィンドウ枠の状態

簡タッチのウィンドウのウィンドウ枠をカスタマイズできます。

◇ 使用関数

`KTC_SetAction(hKTC, KTCA_FRAME, 設定値);`

hKTC : KTC_Create 関数により取得したハンドルを指定します。

設定値 : ウィンドウ枠の設定値を指定します。

◇ 設定値

ウィンドウ枠の設定には、0～3 の数値を設定します。ウィンドウの外観は、タイトルバーのカスタマイズによる設定値との組み合わせにより決定されます。以下には、組み合わせによるウィンドウ枠の状態を示します。

表 5 ウィンドウ枠の状態

		ウィンドウ枠の設定値				
		0	1	2	3	
タイトルバーの設定値 (タイトルバーの有無)	0 (無し)	① 不可	② 不可	③ 可能	③ 不可	←外観番号 ←サイズ変更の可否
	1～5 (有り)	④ 不可	④ 不可	⑤ 可能	④ 不可	←外観番号 ←サイズ変更の可否

■ 外観番号① : タイトルバー無し、枠無し



- 外観番号② : タイトルバー無し、細線の枠



- 外観番号③ : タイトルバー無し、ダイアログ枠



- 外観番号④ : タイトルバー有り、ダイアログ枠



-
- 外観番号⑤ : タイトルバー有り、ダイアログ枠(サイズ変更枠)



※外観④に比べて、枠が少しだけ太い

◇ コーディング例

ウインドウ枠設定として、3 を設定する例を以下に示します。

```
KTC_SetAction( hKTC, KTCA_FRAME, 3 );
```

7.2. タイトルバーの状態

タイトルバーの有無や、アイコン・閉じるボタン・最大化ボタン・最小化ボタンの表示をカスタマイズできます。

◇ 使用関数

```
KTC_SetAction( hKTC, KTCA_TITLEBAR, 設定値 );
```

hKTC : KTC_Create 関数により取得したハンドルを指定します。

設定値 : タイトルバーの設定値を指定します。

◇ 設定値

タイトルバーの設定値は、0～5 の設定が可能です。

表 6 タイトルバーの状態

設定値	キャプション	アイコン	閉じる(×)ボタン	最大化ボタン	最小化ボタン
0	タイトルバーなし				
1	○	×	×	×	×
2	○	○	○	×	×
3	○	○	○	○	▲
4	○	○	○	▲	○
5	○	○	○	○	○

×:表示なし、○:表示あり・使用可能、▲:表示はあるが使用不能

■ 設定値0 : タイトルバー無し



■ 設定値1 : キャプションのみ



■ 設定値2 : キャプション、アイコン、閉じるボタン



-
- 設定値3 : キャプション、アイコン、閉じるボタン、最大化ボタン



- 設定値4 : キャプション、アイコン、閉じるボタン、最小化ボタン



- 設定値5 : キャプション、アイコン、閉じるボタン、最大化ボタン、最小化ボタン



◇ コーディング例

タイトルバーの設定として、5 を設定する例を以下に示します。

```
KTC_SetAction( hKTC, KTCA_TITLEBAR, 5 );
```

7.3. リスト表示キーの表示

リスト表示キーの有無をカスタマイズできます。なお、外部モードの場合は、「有」に設定してもリスト表示キーは表示されません。

◇ 使用関数

```
KTC_SetAction( hKTC, KTCA_LISTBTN, 設定値 );
```

hKTC : KTC_Create 関数により取得したハンドルを指定します。

設定値 : リスト表示キーの有無を指定します。

◇ 設定値

リスト表示キーの有／無を以下の値で設定します。

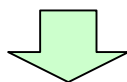
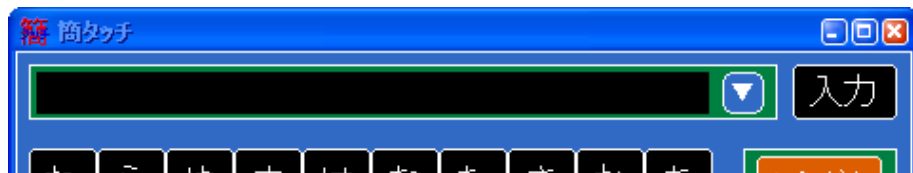
- ・有りの場合 : 1
- ・無しの場合 : 0

◇ コーディング例

リスト表示キー無しを設定する例を以下に示します。

```
KTC_SetAction( hKTC, KTCA_LISTBTN, 0 );
```

<実行結果>



↓
リスト表示キーが
無くなります。



7.4. 入力文字数の制限

内部モード時、文字表示枠に入力できる文字数を制限します。

◇ 使用関数

```
KTC_SetAction( hKTC, KTCA_LIMITTEXT, 設定値 );
```

hKTC : KTC_Create 関数により取得したハンドルを指定します。

設定値 : 最大入力バイト数を指定します。

◇ 設定値

最大入力バイト数として、0～1000 までの値が設定可能です。

0 または範囲外の値の場合は、最大値 1000 が設定されます。

◇ コーディング例

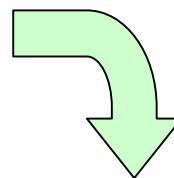
入力バイト数を 10 バイトに制限する例を以下に示します。

```
KTC_SetAction( hKTC, KTCA_LIMITTEXT, 10 );
```

<実行結果>



左図の様に入力して「決定」すると、下図の様に 10 バイトまでしか入力されません。全角の場合は1文字 2 バイトで計算します。



7.5. 外部／内部モードの切り替え

簡タッチを外部モード状態または内部モード状態に設定します。

◇ 使用関数

```
KTC_SetAction( hKTC, KTCA_EXTERNAL, 設定値 );
```

hKTC : KTC_Create 関数により取得したハンドルを指定します。

設定値 : モード示す値を指定します。

◇ 設定値

以下の値を設定します。

- ・内部モード : 0
- ・外部モード : 1
- ・外部モード2 : 2

◇ コーディング例

内部モードに設定する例を以下に示します。

```
KTC_SetAction( hKTC, KTCA_EXTERNAL, 0 );
```



内部モード



外部モード



外部モード2

7.6. ウィンドウ移動の可否

簡タッチはタイトルバーが無い場合でも、キー以外の場所をマウスドラッグしてウィンドウの移動ができるようになっていますが、この機能を ON/OFF することが可能です。

◇ 使用関数

```
KTC_SetAction( hKTC, KTCA_MOVEWINDOW, 設定値 );
```

hKTC : KTC_Create 関数により取得したハンドルを指定します。

設定値 : ウィンドウ移動の ON/OFF を示す値を指定します。

◇ 設定値

以下の値を設定します。

- ON : 1
- OFF : 0

◇ コーディング例

ウィンドウ移動をOFFに設定する例を以下に示します。

```
KTC_SetAction( hKTC, KTCA_MOVEWINDOW, 0 );
```

7.7. IME ON/OFF キーの表示

IME ON/OFF キーの有無をカスタマイズできます。

◇ 使用関数

```
KTC_SetAction( hKTC, KTCA_IMEONOFF, 設定値 );
```

hKTC : KTC_Create 関数により取得したハンドルを指定します。

設定値 : IME ON/OFF キーの有無を指定します。

◇ 設定値

IME ON/OFF キーの有／無を以下の値で設定します。

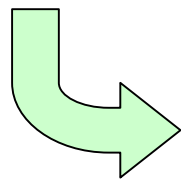
- ・有りの場合 : 1
- ・無しの場合 : 0

◇ コーディング例

IME ON/OFF キー無しを設定する例を以下に示します。

```
KTC_SetAction( hKTC, KTCA_IMEONOFF, 0 );
```

<実行結果>



IME ON/OFF キーが
非表示になった。



7.8. 変換キーの表示

変換キーの有無をカスタマイズできます。

◇ 使用関数

```
KTC_SetAction( hKTC, KTCA_CONVBTN, 設定値 );
```

hKTC : KTC_Create 関数により取得したハンドルを指定します。

設定値 : 変換キーの有無を指定します。

◇ 設定値

変換キーの有／無を以下の値で設定します。

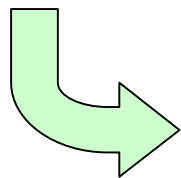
- ・有りの場合 : 1
- ・無しの場合 : 0

◇ コーディング例

変換キー無しを設定する例を以下に示します。

```
KTC_SetAction( hKTC, KTCA_CONVBTN, 0 );
```

<実行結果>



変換キーが非表示になった。↗

7.9. ウィンドウの位置と大きさ

簡タッチのウィンドウの位置と大きさを設定可能です。

◇ 使用関数

```
KTC_SetDlgPosition( hKTC, 設定値 );
```

hKTC : KTC_Create 関数により取得したハンドルを指定します。

設定値 : 位置と大きさを指定します。

◇ 設定値

簡タッチウィンドウの左上の位置と、ウィンドウの大きさ(幅、高さ)を設定します。

位置と大きさは、常にスクリーン座標に対する値を指定します。

簡タッチのクライアントエリアの幅と高さの比率は常に一定です。そのため、幅と高さの内、より小さいウィンドウサイズになる方の値を使い、他方の値は自動調整されます。

幅か高さのどちらかが0の場合は、0でない方の値を使い、他方の値は自動調整されます。

◇ コーディング例

ウィンドウの位置を(50,100)、ウィンドウ幅を 480 になるように設定する例を以下に示します。

```
KTC_POSITION pos;  
pos.X      = 50;          /* X方向の位置 */  
pos.Y      = 100;       /* Y方向の位置 */  
pos.Width  = 480;       /* 幅          */  
pos.Height = 0;         /* 高さ        */  
  
KTC_SetDlgPosition( hKTC, &pos );
```

7.10. ウィンドウのキャプション

簡タッチのウィンドウのキャプションを設定可能です。

◇ 使用関数

```
KTC_SetDlgCaption( hKTC, キャプション文字列 );
```

hKTC : KTC_Create 関数により取得したハンドルを指定します。

キャプション文字列 : キャプションに設定する文字列を指定します。

◇ キャプション文字列

NULL 文字で終わる、マルチバイト文字コード(通常 Shift-JIS コード)を指定します。

◇ デフォルト状態

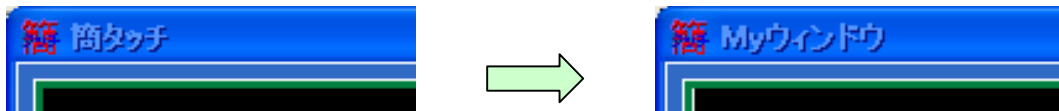
“簡タッチ”という文字列がキャプションになります。

◇ コーディング例

キャプションを“My ウィンドウ”という文字列にする例を以下に示します。

```
KTC_SetDlgCaption( hKTC, “My ウィンドウ” );
```

<実行結果>



7.11. ウィンドウのアイコン

簡タッチウィンドウのアイコンを設定可能です。

◇ 使用関数

```
KTC_SetDlgIcon( hKTC, 設定値 );
```

hKTC : KTC_Create 関数により取得したハンドルを指定します。

設定値 : アイコンを指定します。

◇ 設定値

Win32 の LoadIcon 関数や LoadImage 関数等で取得したハンドルを指定します。

◇ デフォルト状態

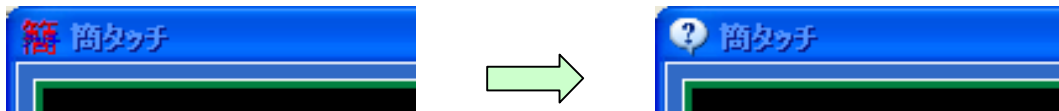
コーディング例の実行前のウィンドウをご覧ください。

◇ コーディング例

Windows であらかじめ定義されている疑問符アイコンを、簡タッチのアイコンに設定する例を示します。

```
HICON hIcon = LoadIcon( NULL, IDI_QUESTION ); // 疑問符アイコンのハンドル取得  
KTC_SetDlgIcon( hKTC, hIcon );
```

<実行結果>



7.12. 文字表示枠への文字列設定

文字表示枠に文字を直接設定することが可能です。

◇ 使用関数

```
KTC_SetText( hKTC, 文字列, 文字列長 );
```

```
KTC_SetText2( hKTC, 文字列, カーソル位置 );
```

hKTC : KTC_Create 関数により取得したハンドルを指定します。

文字列 : 設定する文字列を指定します。

文字列長 : 文字列のバイト数を指定します。0を指定すると NULL 終端までを設定します。

カーソル位置 : カーソル位置を指定します。

◇ 文字列

NULL 文字で終わる、マルチバイト文字コード(通常 Shift-JIS コード)を指定します。

◇ カーソル位置

たとえば、1を指定すると1バイト目の後ろにカーソルが設定されます。カーソル位置が全角文字の真ん中を指す場合は、その文字の前にカーソルが設定されます。

最前部の場合は 0、負の値や文字列のバイト数以上の値を指定すると最後尾にカーソルが設定されます。

◇ デフォルト状態

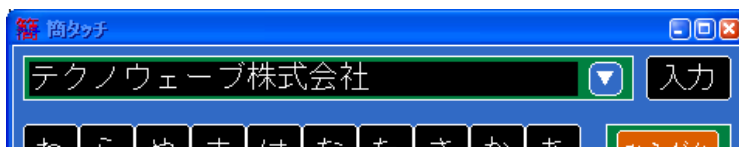
簡タッチの起動直後は、文字表示枠に何も表示されていません。

◇ コーディング例

文字表示枠に“テクノウェーブ株式会社”という文字列を設定する例を以下に示します。

```
KTC_SetText( hKTC, “テクノウェーブ株式会社” );
```

<実行結果>



7.13. リスト設定

リスト表示画面に表示されるリストを設定することが可能です。

◇ 使用関数

```
KTC_SetList( hKTC, リスト文字列, 文字列長 );
```

hKTC : KTC_Create 関数により取得したハンドルを指定します。

リスト文字列 : リストとする文字列を指定します。

文字列長 : リスト文字列の長さ、もしくはそれ以上の値を指定します。

リスト文字列以上の値を指定した場合は、NULL が2個続く所まで読み取ります。

◇ リスト文字列

リスト文字列は、リストとして表示する先頭行の文字列から最終行までの文字列を連結して、さらに最後に NULL を1個追加したものです。各行の文字列は NULL が最後に付きます。

リスト文字列は、マルチバイト文字コード(通常 Shift-JIS コード)です。

◇ デフォルト値

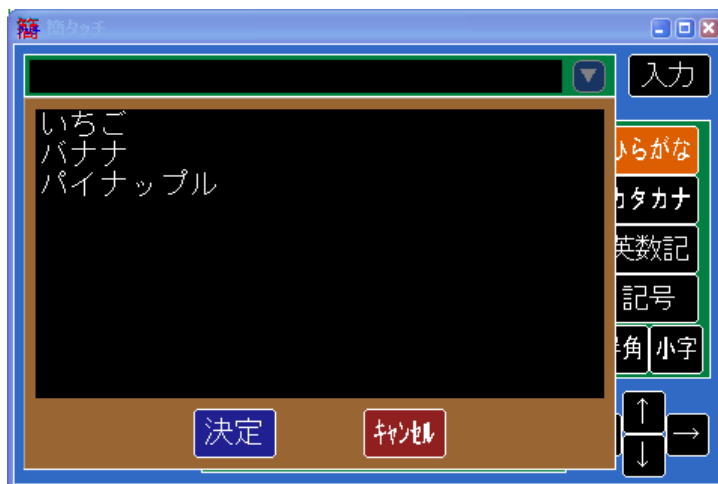
デフォルトではリストとして何も設定がされていません。

◇ コーディング例

リストを設定する例を以下に示します。

```
char list[1000] = "いちご¥0 バナナ¥0 パイナップル¥0"  
KTC_SetList( hKTC, list, 1000 );
```

<実行結果>



7.14. キャプション

簡タッチのキーのキャプションを変更することが可能です。

◇ 使用関数

```
KTC_SetCaption( hKTC, オブジェクト番号, メインテーブル番号, サブテーブル番号,  
               キャプション文字列 );
```

hKTC : KTC_Create 関数により取得したハンドルを指定します。

オブジェクト番号 : 設定対象キーのオブジェクト番号を指定します。

メインテーブル番号 : 設定対象のメインテーブル番号を指定します。

サブテーブル番号 : 設定対象のサブテーブル番号を指定します。

キャプション文字列 : 設定するキャプションを指定します。

※オブジェクト番号については、「7.18 オブジェクト番号について」をご覧ください。

※メインテーブル番号とサブテーブル番号については、後述します。

◇ キャプション文字列

キャプション文字列は、NULL 文字で終わるマルチバイト文字コード(通常 Shift-JIS コード)です。
また、文字入力キーに設定できる文字列長は最大4バイト(NULL の終端文字の1バイトを含まず)で、その他のキーに設定できる文字列長は最大12バイト(NULL の終端文字の1バイトを含まず)です。

空キャプション : キャプション文字として空文字("")を指定した場合は表示が下図のようになり、使用不可(クリックを受け付けない)になります。



図 9 空キャプション表示例

◇ テーブル番号

メインテーブル番号とサブテーブル番号は、テーブル選択キーの状態を表す数値です。下図のように、メインテーブル番号はメインテーブル選択キー(上4つのキー)の状態を示し、サブテーブル番号はサブテーブル選択キー(下の2個のキー、図では「半角」と「小字」のキー)の状態を示します。

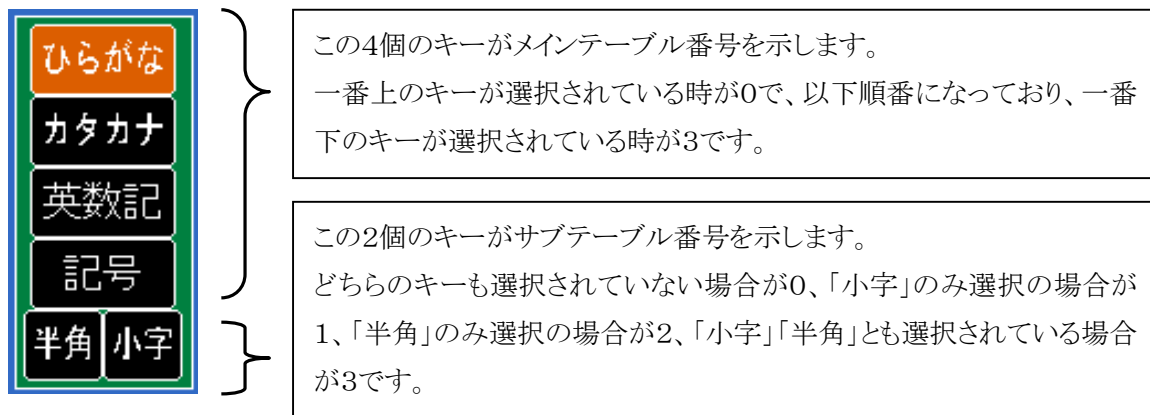


図 10 テーブル番号説明図

オブジェクト番号により、必要になるテーブル番号が変わります。下表をご覧ください。

表 7 必要なテーブル番号

オブジェクト番号 が指すもの	メインテーブル番号 が必要なもの	サブテーブル番号 が必要なもの	空キャプション が機能するもの
文字入力キー	○	○	○
入力通知キー	×	×	○
メインテーブル選択キー	×	×	○
サブテーブル選択キー	○※1	×	○
矢印キー	×	×	○
IMEキー	○※2	×	○
リスト表示キー	×	×	○
リスト表示ウィンドウの 「決定」「キャンセル」キー	×	×	×

※1 サブテーブル選択キーは、メインテーブル番号毎に設定できます。

※2 スペースキーは、メインテーブル番号0が全角時、1が半角時、2が変換時の設定です。

変換キーは、メインテーブル番号0が変換時、1が無変換時の設定です。

必要でないテーブル番号は0を指定して下さい。

◇ コーディング例



左図の画面の赤丸が付いたキーにキャプションを設定する例を示します。

ここにキャプションを設定する

まず、赤丸の示すキーのオブジェクト番号は、52です。

メインテーブル選択キーは上から3番目の「英数記」が選択されているので、メインテーブル番号は2です。サブテーブル選択キーは「半角」「小字」の両方選択されているので、サブテーブル番号は3です。

ここでは、キャプションに全角の”?”を設定してみると、以下のコーディングとなります。

```
KTC_SetCaption( hKTC, 52, 2, 3, " ? " ); // キャプション設定
KTC_UpdateView( hKTC ); // 画面更新
```

実行結果は、下図のとおりです。



ここが ? に変わった

7.15. フォント

キャプションで設定した文字のフォントを変更することが可能です。

◇ 使用関数

`KTC_SetFont(hKTC, オブジェクト番号, メインテーブル番号, サブテーブル番号, フォント情報);`

hKTC : KTC_Create 関数により取得したハンドルを指定します。

オブジェクト番号 : 設定対象キーのオブジェクト番号を指定します。

メインテーブル番号 : 設定対象のメインテーブル番号を指定します。

サブテーブル番号 : 設定対象のサブテーブル番号を指定します。

フォント情報 : フォントに関する情報を指定します。

※オブジェクト番号については、「7.18 オブジェクト番号について」をご覧ください。

※メインテーブル番号とサブテーブル番号については、「7.14 キャプション」をご覧ください。

◇ フォント情報

フォントの高さ、幅、太さ、フォント名を指定します。

高さは、値が大きいほど大きなフォントになります。

幅は、値が小さいほど幅の細いフォントになります。0を指定するとデフォルトの幅になります。デフォルトのフォントの場合、14がデフォルト幅です。

太さは、小さい値ほど線の細いフォントになります。0を指定するとデフォルトの太さになります。400が標準の太さ、通常の太字は700です。

※フォント情報はデフォルトの情報をもとに、カスタマイズするのが簡単です。

コーディング例は次頁をご覧ください。

◇ コーディング例

「7.14 キャプション」のコーディング例で設定したキャプションのフォントを変更する例を示します。
ここでは、KTC_GetFont 関数により現在のフォントを読み出し、高さを変えてみます。

```
char buf[100]; // フォント名を格納するバッファ
KTC_FONT font; // フォント情報用変数

font.pFaceName = buf; // バッファのポインタを設定
font.len = sizeof(buf); // バッファの大きさを設定
KTC_GetFont( hKTC, 52, 2, 3, &font ); // フォント情報取得

font.height = 350; // フォントの高さ変更
KTC_SetFont( hKTC, 52, 2, 3, &font ); // フォント情報を設定

KTC_UpdateView( hKTC ); // 画面更新
```

<実行結果>



文字が大きくなった

7.16. 色

画面上の各部の色を変更することが可能です。

◇ 使用関数

`KTC_SetColor(hKTC, オブジェクト番号, カラー情報0, カラー情報1, カラー情報2);`

`hKTC` : KTC_Create 関数により取得したハンドルを指定します。

`オブジェクト番号` : 設定対象のオブジェクト番号を指定します。

`カラー情報0~2` : カラーに関する情報を指定します。

※オブジェクト番号については、「7.18 オブジェクト番号について」をご覧ください。

※色に関しては、テーブル番号による設定がありません。(テーブル間で共通です)

◇ カラー情報

カラー情報は0~2の3つがあり、1つのカラー情報は変数型 `KTC_COLOR` として、1つのオブジェクトが表示するのに必要な情報全てが入っています。(`KTC_COLOR` の詳細は「10.2.7 `KTC_COLOR` 型」をご覧ください)

3つのカラー情報は、通常表示の場合、空キャプションのキー表示の場合、選択された時の場合等で使い分けられます。下表にオブジェクト種類とカラー情報の使い分けを示します。

表 8 カラー情報の使い分け

オブジェクト種類	カラー情報0	カラー情報1	カラー情報2
文字入力キー	通常が表示の時	空キャプションの時	(未使用)
入力通知キー	通常が表示の時	空キャプションの時	(未使用)
テーブル選択キー	通常が表示の時	空キャプションの時	選択時
矢印キー	通常が表示の時	空キャプションの時	(未使用)
IMEキー	通常が表示の時	空キャプションの時	変換時
リスト表示キー	リスト未表示の時	リスト表示中の時	(未使用)
その他	通常が表示の時	(未使用)	(未使用)

コーディング例は次頁をご覧ください。

◇ コーディング例

「7.14 キャプション」のコーディング例で設定したキーの色を変更する例を示します。
ここでは、KTC_GetColor 関数により現在のカラーを読み出し、背景色を変えてみます。

```
KTC_COLOR color0, color1, color2; // カラー情報用変数
KTC_GetColor( hKTC, 52, &color0, &color1, &color2 ); // カラー情報取得

color0.back = RGB( 255, 0, 255 ); // 背景色変更
KTC_SetColor( hKTC, 52, &color0, &color1, &color2 ); // カラー情報設定

KTC_UpdateView( hKTC ); // 画面更新
```

<実行結果>



背景色が変わった

7.17. 文字コード

文字入力キーに対応付けられている文字コードを変更することが可能です。

◇ 使用関数

`KTC_SetKeyCode(hKTC, オブジェクト番号, メインテーブル番号, サブテーブル番号, 文字コード, IME モード);`

hKTC : KTC_Create 関数により取得したハンドルを指定します。

オブジェクト番号 : 設定対象キーのオブジェクト番号を指定します。

メインテーブル番号 : 設定対象のメインテーブル番号を指定します。

サブテーブル番号 : 設定対象のサブテーブル番号を指定します。

文字コード : キーをクリックしたときに入力される文字列を指定します。

IME モード : 文字コードを入力する前に設定しておくIMEのモードを指定します。

※オブジェクト番号については、「7.18 オブジェクト番号について」をご覧ください。

※メインテーブル番号とサブテーブル番号については、「7.14 キャプション」をご覧ください。

◇ 文字コード

文字コードは、NULL 文字で終わる文字列(通常 Shift-JIS コード)で、最大文字列長は6バイト (NULL の終端文字の1バイトを含まず)です。

◇ IME モード

下表にIMEモードの一覧を示します。

表 9 IME モード値① 入力モード設定用定義

IME モード	IME の設定状態		値		備考
	入力モード	変換状態	10 進	16 進	
KTCIME_HAN_EISU	半角英数	変換	0	0h	
KTCIME_HAN_KATA	半角カタカナ	変換	1	1h	
KTCIME_ZEN_EISU	全角英数	変換	2	2h	
KTCIME_ZEN_KATA	全角カタカナ	変換	3	3h	
KTCIME_ZEN_HIRA	全角ひらがな	変換	7	7h	

表 10 IME モード値② 入力モードとの組み合わせ用定義

IME モード	IME の設定状態		値		備考
	入力モード	変換状態	10 進	16 進	
KTCIME_NOCONV	—	無変換	8	8h	無変換モードにする
KTCIME_OFF	OFF	無変換	32	20h	IME を OFF にする
KTCIME_SHIFT	—	—	4	4h	大文字入力にする
KTCIME_ROMAN	—	—	16	10h	ローマ字入力にする

IME モード②を使用しない場合のデフォルト状態は、文字コードはマルチバイト文字(全角と半角文字の混在)、IME は ON で変換状態となります。また、KTCIME_ROMAN を使用した英数モードの場合、小文字の英字が入力されます。

IME モード値②は、このデフォルト状態を変更するために、IME モード①および IME モード②と組み合わせ(論理 OR)で使用します。

KTCIME_NOCONV は、IME を無変換モードに設定します。

KTCIME_OFF は、IME OFF(または入力モードを直接入力)にします。

KTCIME_ROMAN は、IME モードをローマ字漢字変換に設定します。文字コードは半角大文字英字(A~Z)と半角数字(0~9)のみ可能です。

KTCIME_SHIFT は、KTCIME_ROMAN を使用した英数モード時に、大文字英字が入力されるようにします。

コーディング例は次頁をご覧ください。

◇ コーディング例

「7.14 キャプション」のコーディング例で設定したキーに実際に入力されるコードを設定する例を示します。

```
KTC_SetKeyCode( hKTC, 52, 2, 3, "きゃ", KTCIME_ZEN_HIRA ); // 文字コード設定
```

<実行結果>



■ MS IME での注意事項

MS IME(IME-2002 など)を使用する場合、無変換文字として入力したい場合は KTCIME_NOCONV を使用してください。KTCIME_OFF では正しく動作しない場合があります。

■ ATOK での注意事項

MS IME のようにマルチバイト文字の入力が正しく動作しません。(ATOK15 で確認済み)
KTCIME_OFF ではマルチバイト文字を正しく入力できます。ただし、確定入力となります。
変換文字として入力したい場合は、KTCIME_ROMAN を使用して下さい。

以下は、文字“きゃ”を入力するコーディング例です。

```
KTC_SetKeyCode( hKTC, 52, 2, 3, "KYA", KTCIME_ROMAN | KTCIME_ZEN_HIRA );
```

7.18. オブジェクト番号について

オブジェクト番号は、キャプション、フォント、色、文字コードの設定先を示す値です。

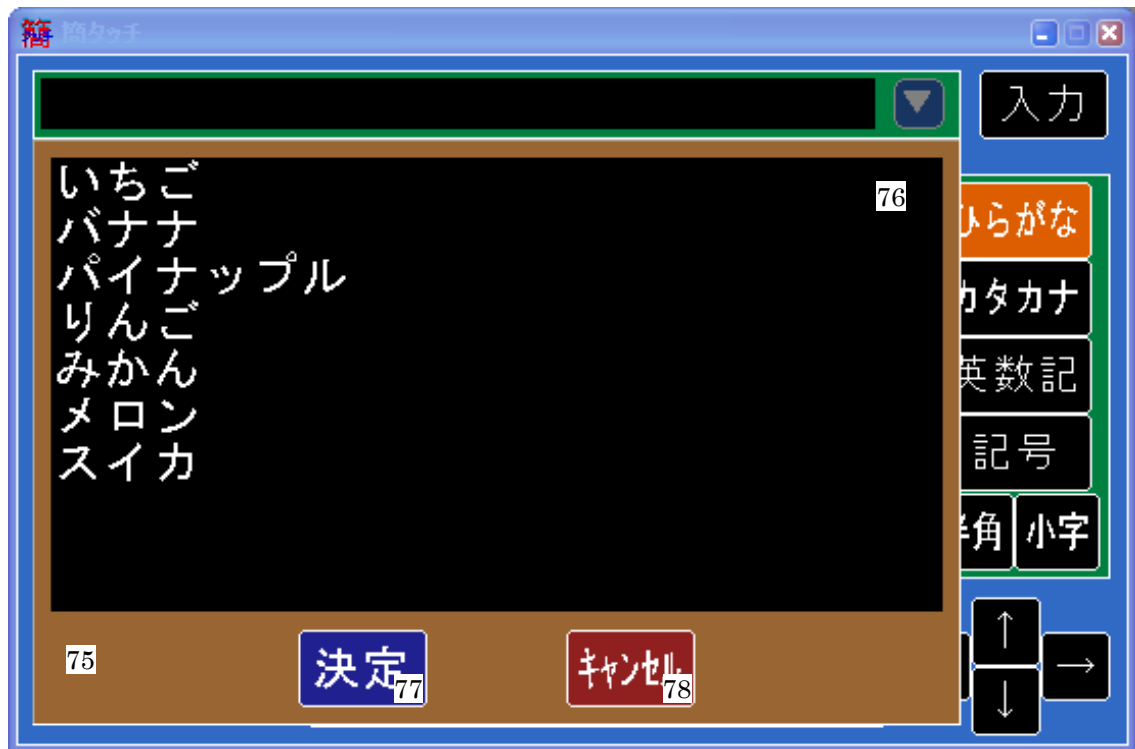
以下の簡タッチ画面のキー等に記載されている番号がオブジェクト番号です。

◇ メイン画面



オブジェクト番号	対象物
0～52	文字入力キー
53	入力通知キー
54～59	テーブル選択キー
60～63	矢印キー
64～66	IME キー 64:後退キー、65:スペースキー、66:Enter キー
67	リスト表示キー
68	テーブル選択キーを囲む枠
69	IME キーを囲む枠
70	文字表示枠を囲む枠
71	簡タッチウィンドウのクライアントエリアを囲む枠
72	文字表示枠
73	簡タッチウィンドウのクライアントエリア
74	IME ON/OFF キー
79	変換キー

◇ リスト表示画面



オブジェクト番号	対象物
75	リスト表示画面のクライアントエリア
76	リスト欄
77	決定キー
78	キャンセルキー

7.19. デフォルトの設定について

簡タッチのデフォルト設定状態を以下に示します。

表 11 デフォルト設定

設定対象	設定値	備考
ウィンドウ枠	2	タイトルバー有り、最大化・最小化ボタン使用可能、 ウィンドウサイズ変更可能
タイトルバー	5	
リスト表示キー	有り	
IME ON/OFF キー	有り	
変換キー	有り	
入力文字数制限	1000	
外部／内部モード	外部モード	
ウィンドウ移動	可能	
ウィンドウの位置と大きさ	クライアント領域の幅が 640になるように設定	位置はOSが決定。
ウィンドウのキャプション	”簡タッチ”	
ウィンドウのアイコン	専用アイコン	
文字表示枠の文字	なし	
リスト表示の文字列	なし	
キャプション	—	(簡タッチ画面でご確認ください)
フォント	(後述)	フォント名はすべてMSゴシックを使用
色	(後述)	
文字コード	—	キャプションと同じものを設定。

◇ フォントのデフォルト値

表 12 フォントのデフォルト設定

対象物	高さ	幅	太さ	フォント名
リスト表示キー	130	0	400	MS ゴシック
IME ON/OFF キー	170	0	400	MS ゴシック
入力通知キー、テーブル選択キー、 矢印キー、スペースキー	190	0 または 10	400	MS ゴシック
文字入力キー、後退キー、 決定キー、文字表示枠、 リスト表示画面のリスト欄と 決定・キャンセルキー	210	0 または10 または12	400	MS ゴシック

◇ 色のデフォルト値

表 13 色のデフォルト設定

対象物	オブジェクト 番号	カラー情報 の番号	frame	back	text	cursor
文字入力キー 入力通知キー 矢印キー	0~52 53 60~63	0 (通常表示)	255,255,255	0,0,0	255,255,255	229,151,0
		1 (空キャプション)	128,128,128	50,50,50	---	50,50,50
		外部モード2の時の 閉じるキー (入力通知キー)	53	0 (通常表示)	255,255,255	194,23,29
		1 (空キャプション)	128,128,128	50,50,50	---	50,50,50
		2 (選択時)	255,255,255	220,94,0	255,255,255	229,151,0
テーブル選択キー	54~59	0 (通常表示)	255,255,255	0,0,0	255,255,255	229,151,0
		1 (空キャプション)	128,128,128	50,50,50	---	50,50,50
		2 (変換時)	255,255,255	0,0,0	255,255,255	229,151,0
IMEキー (後退キー、 スペースキー、 Enter キー、 変換キー)	64~66 79	0 (通常表示)	255,255,255	0,0,0	255,255,255	229,151,0
		1 (空キャプション)	128,128,128	50,50,50	---	50,50,50
		2 (変換時)	255,255,255	0,0,0	255,255,255	229,151,0
IME ON/OFF キー	74	0 (通常表示)	255,255,255	128,0,0	255,255,255	229,151,0
		1 (空キャプション)	128,128,128	50,50,50	---	50,50,50
		2 (変換時)	255,255,255	128,0,0	255,255,255	229,151,0
リスト表示キー	67	0 (リスト未表示)	255,255,255	49,106,197	255,255,255	229,151,0
		1 (リスト表示中)	128,128,128	24,53,98	128,128,128	---
テーブル選択キーを囲む枠	68	0	255,255,255	0,128,64	---	---
IMEキーを囲む枠	69					
文字表示枠を囲む枠	70					
簡タッチウィンドウの クライアントエリアを囲む枠	71	0	255,255,255	49,106,197	---	---
文字表示枠	72	0	---	0,0,0	255,255,255	---
簡タッチウィンドウの クライアントエリア	73	0	---	0,0,0	---	---
リスト表示画面の クライアントエリア	75	0	---	153,102,51	---	---
リスト欄	76	0	---	0,0,0	255,255,255	---
リスト表示画面の 決定キー	77	0	255,255,255	32,32,144	255,255,255	229,151,0
リスト表示画面の キャンセルキー	78	0	255,255,255	144,32,32	255,255,255	229,151,0

欄内の色は、RGB の順で値を示しています。欄内の---は使用されない項目を示しています。

8. イベント通知機能

簡タッチにはアプリケーションへ簡タッチ上のイベントを知らせる、イベント通知機能があります。
この機能によりアプリケーションは適切なタイミングで、簡タッチに対して状態確認や指示をすることが可能となります。

8.1. 通知方式

通知方式には、メッセージ方式とコールバック方式があり、どちらの場合も `KTC_SetNotify` 関数を使って設定します。

◇メッセージ方式の場合

以下の Win32API のメッセージ関数を用いて、アプリケーションへイベント通知がされます。

```
PostMessage(hWnd, nMsgNo, (WPARAM)reason, (LPARAM)lParam)  
または、
```

```
SendMessage(hWnd, nMsgNo, (WPARAM)reason, (LPARAM)lParam)
```

ウィンドウハンドル `hWnd` およびメッセージ番号 `nMsgNo` の値は、`KTC_SetNotify` 関数で設定した値です。`reason` は通知理由を示す値です。`lParam` は通知理由に固有の値が設定されます。`PostMessage` と `SendMessage` のどちらで通知されるかは、通知理由により決まります。

◇コールバック方式の場合

関数呼び出しで、アプリケーションへイベント通知されます。この方式は、アプリケーションがウィンドウを持たない場合に有効です。コールバック関数は以下の形式です。

```
long (stdcall *fCallback)(UINT nMsgNo, KTC_ENREASON reason, long lParam)
```

`fCallback` および `nMsgNo` の値は `KTC_SetNotify` 関数で設定した値です。`reason`, `lParam` の意味はメッセージ方式と同様です。

※メッセージ方式、コールバック方式のいずれの場合も、その処理ハンドラの中で簡タッチのインスタンス削除関数 (`KTC_Delete` 関数) を呼び出すことはできません。

8.2. 通知理由

通知理由は KTC_ENREASON 型として定義されており、下表に示すものがあります。

表 14 通知理由

通知理由 reason	意 味	IParam の意味	メッセージ方式の場合 の呼び出し関数
KTCR_BTNINPUT	入力通知キーが押された	(未使用)	SendMessage
KTCR_BTNCLOSE	タイトルバーの×ボタンが押された	(未使用)	SendMessage
KTCR_MOVEWINDOW	ウインドウの位置・サイズが変わった	(未使用)	PostMessage
KTCR_LISTOPEN	リスト表示キーが押された	(未使用)	SendMessage
KTCR_TABLECHANGE	テーブル選択キーが押された	テーブル番号	SendMessage
KTCR_HIDE	ウインドウが非表示になろうとしている。	(未使用)	SendMessage
KTCR_BKCLICK	キー等をクリックした。	キー番号	SendMessage
KTCR_UKCLICK	ユーザー定義キーをクリックした。 (現在未使用)	ユーザー定義 キー番号	SendMessage
KTCR_IMESTATUS	IME の状態が変化した。	変換状態	PostMessage

◇KTCR_BTNINPUT

入力通知キーが押されると、このイベント通知が発生します。アプリケーションは文字表示枠に表示されている文字列を引き取るタイミングとして利用できます。アプリケーションから戻ると簡タッチが非表示になりますが、戻り値によりこの動作をキャンセルできます。

◇KTCR_BTNCLOSE

タイトルバーの右端にある×(閉じる)ボタンが押されると、このイベント通知が発生します。アプリケーションから戻ると簡タッチが非表示になりますが、戻り値によりこの動作をキャンセルできます。

◇KTCR_MOVEWINDOW

簡タッチのウインドウの移動やサイズ変化が生じた場合に、このイベント通知が発生します。現在のウインドウの位置・サイズは、KTC_GetDlgPositon 関数により取得できます。

◇KTCR_LISTOPEN

リスト表示キーが押されると、このイベント通知が発生します。リスト表示画面はまだ表示していません。したがって、アプリケーションはハンドラ内で KTC_SetList 関数を使用してリスト設定を行うことができます。アプリケーションから戻るとリスト表示画面が表示されますが、戻り値によりこの動作をキャンセルできます。

◇KTCR_IMESTATUS

IME の変換状態が変化した場合に、このイベントが発生します。IParam は変換状態を示しており、0 の場合は変換中でないことを示し、1 の場合は変換中であることを示します。

8.3. アプリケーションからの戻り値による動作

イベント通知用のメッセージハンドラやコールバック関数からの戻り値により、簡タッチの動作を制御することが可能です。下表に、戻り値による簡タッチの動作を示します。

表 15 戻り値による動作

通知理由※1	戻り値※2	戻り値による簡タッチの動作
KTCR_BTNINPUT	0	ウィンドウを非表示にする
	1	ウィンドウを非表示にしない
KTCR_BTNCLOSE	0	ウィンドウを非表示にする
	1	ウィンドウを非表示にしない
KTCR_MOVEWINDOW	0	何もしない
KTCR_LISTOPEN	0	リストを表示する
	1	リストを表示しない
KTCR_TABLECHANGE	0	テーブルを変更する
	1	テーブルを変更しない
	上位ワードが 0xFFFF	下位ワードで指定されるテーブル番号にする
KTCR_HIDE	0	非表示動作を行う
	1	非表示動作をしない
KTCR_BKCLICK	0	文字コードを送信する。
	1	文字コードの送信をしない
KTCR_IMESTATUS	0	何もしない

※1 この表以外の通知理由で呼び出された場合、戻り値は0として下さい。

※2 表にある戻り値以外の値は予約であるため使用しないで下さい。

◇KTC_TABLECHANGE のテーブル番号指定

IParam が示すテーブル番号の上位ワードを 0xFFFF にし、下位ワードの値を変更したものを戻り値とすることで希望のテーブル番号に設定することができます。不正なテーブル番号を設定した場合はテーブルの変更をしません。(「図 11 IParam が示すテーブル番号の構造」を参照)

◇使用上の注意

コールバック方式の場合の通知、およびメッセージ方式で SendMessage による通知は、アプリケーションから戻るまでは簡タッチの動作が停止しますのでご注意ください。

9. 状態の記録と復帰

簡タッチの状態の記録および復帰が可能です。状態を記録するには KTC_Save 関数、記録した状態に復帰するには KTC_Restore 関数を使用します。

9.1. 記録範囲

下表は、保存できる範囲を示しています。

表 16 状態記録範囲

No.	記録内容	KTC_Save()、 KTC_Restore()で範囲指定 する場合のパラメータ※1	簡タッチの設定に 使用する関数
1	ウィンドウ枠	KTCSV_ACTION	KTC_SetAction
2	タイトルバー		
3	リスト表示キー		
4	入力文字数制限		
5	外部／内部モード切り替え		
6	ウィンドウ移動		
7	ダイアログの位置と大きさ	KTCSV_DIALOG	KTC_SetDlgPosition
8	リスト設定	KTCSV_LIST	KTC_SetList
9	キャプション	KTCSV_CAPTION	KTC_SetCaption
10	フォント	KTCSV_FONT	KTC_SetFont
11	色	KTCSV_COLOR	KTC_SetColor
12	文字コード	KTCSV_KEYCODE	KTC_SetKeyCode
13	テーブル番号	KTCSV_TABLE	KTC_SetTableNum

※1:範囲指定は、それぞれを論理ORすることで複数範囲を指定することができます。全の範囲を指定する場合は、KTCSV_ALL を指定します。

記録範囲を KTC_Save 関数のパラメータで指定します。KTCSV_CAPTION、KTCSV_FONT、KTCSV_COLOR、KTCSV_KEYCODE および KTCSV_ALL を指定した場合、書き込みデータ量が多いため書き込みに時間がかかりますのでご注意ください。(通常数秒～十数秒程度で、書き込み先媒体等のシステム条件で変わります)

KTC_Restore 関数による状態復帰も同様に読み出す範囲をパラメータにより指定することが可能です。

9.2. 記録先について

記録では KTC_Save 関数で指定したファイル名が作成され、記録データが書き込まれます。特にパスを指定しなかった場合、Windows ディレクトリにファイルが作成されます。指定したファイル名が既に保存先に存在する場合は上書きされますのでご注意ください。

◇ アプリケーションフォルダの利用

記録先として、アプリケーションデータフォルダを利用すると便利です。

このフォルダは、OS を C ドライブにインストールした場合は通常以下の場所です。

ユーザーごとに記録する場合 : C:\Documents and Settings\ユーザー名\ApplicationData

ユーザー共通で記録する場合 : C:\Documents and Settings\All Users\ApplicationData

ただし、ユーザー共通のフォルダは書込みできるユーザーが制限されていますので注意が必要です。(Administrators グループや Power Users グループに属するユーザーは書込み権限を持っています)

アプリケーションデータの保存の際には、このフォルダ下に会社名のフォルダを作り、さらにその下にアプリケーション名のフォルダを作り、その中に保存する方法が一般的です。

これらのフォルダへのフルパス名をアプリケーションで取得するために、Win32 の SHGetFolderPath 関数を利用できます。

この他に、特殊な方法として保存先をレジストリにリダイレクトする方法があります。

この方法については、次頁に記載します。

◇ レジストリへのリダイレクト

下記に示すリダイレクト機能を用いると、アプリケーションは一切変更せずに保存先をレジストリにすることができます。※この方法はレジストリを直接操作することになるので、慎重に行ってください。

例として、ファイル名 C:¥MyApp¥MyApp.ini への保存をレジストリにリダイレクトさせます。
保存先は HKEY_CURRENT_USER¥Software¥MyApp¥MyData とした場合、以下のようになります。

- ① レジストリをレジストリエディタ(regedit.exe など)で開きます。
- ② HKEY_LOCAL_MACHINE の下の SOFTWARE¥Microsoft¥Windows NT¥CurrentVersion¥IniFileMapping の下に、MyApp.ini という名前のキーを作成します。
SOFTWARE¥Microsoft¥Windows NT¥CurrentVersion¥IniFileMapping の下に、MyApp.ini という名前のキーを作成します。
- ③ そのキーの名前無し値(regedit.exe で見ると、(既定)という名前の値)のデータに、保存先である `USR:Software¥MyApp¥MyData` の文字列を書き込みます。(下図)ここで `USR:`は、`HKEY_CURRENT_USER¥` を表します。
`SYS:`とした場合は、`HKEY_LOCAL_MACHINE¥SOFTWARE¥` を表します。



図 12 MyApp キー作成とデータ設定した所

- ④ HKEY_CURRENT_USER¥SoftWave の下に、MyApp という名前のキーを作成します。
さらに MyApp のキーの下に、MyData という名前のキーを作成します。ただし、一番下のキーについては自動作成されるので、MyData の作成は省略することが可能です。

以上でレジストリへのリダイレクトが完了します。

上記の例では、HKEY_CURRENT_USER にリダイレクトしたため、ユーザー毎に簡タッチの状態記録が可能です。ただし、ユーザー毎に手順④のキー作成の作業を行ってください。

レジストリへのリダイレクトを停止する場合は、③で書き込んだ名前無し値へのデータを空にしてください。(MyApp.ini のキーは、削除しないで下さい)

10. 関数

簡タッチが公開している関数と、関数で使用される変数型定義を以下に説明します。Windows 側で定義されているものを除いて、全て簡タッチ用ヘッダーファイル (Ktouch.h) 内で定義されています。

10.1. 関数一覧

簡タッチが公開している関数の一覧を下記に示します。

表 17 関数一覧

関数名		機能
設定・生成系	取得・削除系	
KTC_Create	KTC_Delete	簡タッチのインスタンスを生成・削除します
KTC_GenWindow KTC_GenWindow2		ウィンドウを生成します
KTC_Show		ウィンドウの表示状態を変更します
KTC_SetText, KTC_SetText2	KTC_GetText	文字表示枠に文字列を設定・取得します
KTC_SetNotify		通知機能の設定を行います
KTC_SetAction	KTC_GetAction	様々な動作の設定・取得をします
KTC_SetDlgPosition	KTC_GetDlgPosition	ダイアログの位置・大きさを設定・取得します
KTC_SetDlgCaption		ダイアログのキャプションを設定します
KTC_SetDlgIcon		ダイアログのアイコンを設定します
KTC_SetList		リスト設定を行います
KTC_UpdateView		画面を更新します
KTC_SetKeyCode	KTC_GetKeyCode	キーの文字コードを設定・取得します
KTC_SetCaption	KTC_GetCaption	キーのキャプションを設定・取得します
KTC_SetFont	KTC_GetFont	キーのキャプションのフォントを設定・取得します
KTC_SetColor	KTC_GetColor	キーと画面全体の色を設定・取得します
KTC_Save	KTC_Restore	状態を保存・復帰します
KTC_GetWindowHandle		簡タッチウィンドウへのハンドルを取得します。
KTC_SetTableNum	KTC_GetTableNum	テーブル番号を設定・取得します。

10.2. 変数型定義

10.2.1. KTC_ERR 型

ほとんどの関数は実行結果を示す KTC_ERR 型の戻り値を返します。

KTC_ERR 型変数は、正常終了が 0、異常終了が負の値の列挙型として定義されています。

表 18 KTC_ERR 型の定義

定義されている名前	意味	値
KTCERR_OK	正常終了	0
KTCERR_FAILED	異常終了:実行エラー	-1
KTCERR_INVALID_HANDLE	異常終了:ハンドルが無効	-2
KTCERR_INVALID_PARAMETER	異常終了:引数が無効	-3

10.2.2. KTC_ENSHOW 型

KTC_Show 関数で使用される、ウィンドウの表示状態を指定する列挙型です。

表 19 KTC_ENSHOW 型の定義

定義されている名前	意味	値
KTCSW_NORMAL	ウィンドウをアクティブにして表示します。ウィンドウが最小化または最大化されていた場合は、その位置とサイズを元に戻します。	0
KTCSW_MINIMIZE	ウィンドウを最小化します。	1
KTCSW_MAXIMIZE	ウィンドウを最大化します。	2
KTCSW_HIDE	ウィンドウを非表示にします。	3

10.2.3. KTC_ENREASON 型

簡タッチからアプリケーションに通知されたイベントの理由を示す列挙型です。

表 20 KTC_ENSHOW 型の定義

定義されている名前	意味	値
KTCR_BTNINPUT	入力通知キーが押された	0
KTCR_BTNCLOSE	タイトルバーの×ボタンが押された	1
KTCR_MOVEWINDOW	ウィンドウの位置・大きさが変わった	2
KTCR_LISTOPEN	リスト表示キーが押された	3
KTCR_TABLECHANGE	テーブル選択キーが押された	4
KTCR_HIDE	ウィンドウが非表示になろうとしている	5
KTCR_BKCLICK	キー等をクリックした	6
KTCR_UKCLICK	ユーザー定義キーをクリックした	7

KTCR_IMESTATUS	IME の状態が変化した	8
----------------	--------------	---

10.2.4. KTC_ENACTION 型

KTC_SetAction 関数、KTC_GetAction 関数で使用される動作項目を示す列挙型です。

表 21 KTC_ENACTION 型の定義

定義されている名前	意味	値
KTCA_FRAME	ウィンドウ枠の種類や有無	0
KTCA_TITLEBAR	タイトルバーの有無や最大化・最小化ボタンの有効／無効	1
KTCA_EXTERNAL	リスト表示キーの有無	2
KTCA_LISTBTN	文字表示枠に入力できる文字数の制限	3
KTCA_LIMITTEXT	外部／内部モード切り替え	4
KTCA_MOVEWINDOW	クライアント領域でのマウスドラッグでウィンドウ移動の可否	5

10.2.5. KTC_POSITION 型

KTC_SetDlgPosition 関数、KTC_GetDlgPosition 関数で使用される、ウィンドウの位置と大きさを示す構造体です。

```
typedef struct tagKTC_POSITION {
    long X;
    long Y;
    long Width;
    long Height;
} KTC_POSITION;
```

メンバ

X、Y

ダイアログの左上(Xは左、Yは上)の座標を示します。

Width

ダイアログの幅を示します。

Height

ダイアログの高さを示します。

10.2.6. KTC_FONT 型

KTC_SetFont 関数、KTC_GetFont 関数で 사용되는、フォント情報を示す構造体です。

```
typedef struct tagKTC_FONT {  
    long height;  
    long width;  
    long weight;  
    char *pFaceName;  
    long len;  
} KTC_FONT;
```

メンバ

height

フォントの高さを示します。

0以上の値で、大きいほど大きな文字になります。

width

フォントの幅を示します。

0以上の値で、大きいほど幅が広がります。デフォルトの幅に設定するには0を指定します。

weight

フォントの太さを示します。

0以上の値で、大きいほど太くなります。通常太さは 400、太字は 700 程度になります。

pFaceName

フォント名格納バッファへのポインタを指定します。

len

pFaceName のバッファの大きさ(バイト長)を示します。

この値は KTC_GetFont 関数に対して必要で、KTC_SetFont 関数では未使用です。

10.2.7. KTC_COLOR 型

KTC_SetColor 関数、KTC_GetColor 関数で使用する、カラー情報を示す構造体です。

```
typedef struct tagKTC_COLOR {  
    COLORREF frame;  
    COLORREF back;  
    COLORREF text;  
    COLORREF cursor;  
} KTC_COLOR;
```

メンバ

frame, back, text, cursor

対象オブジェクトの各部の色を示します。(下図)

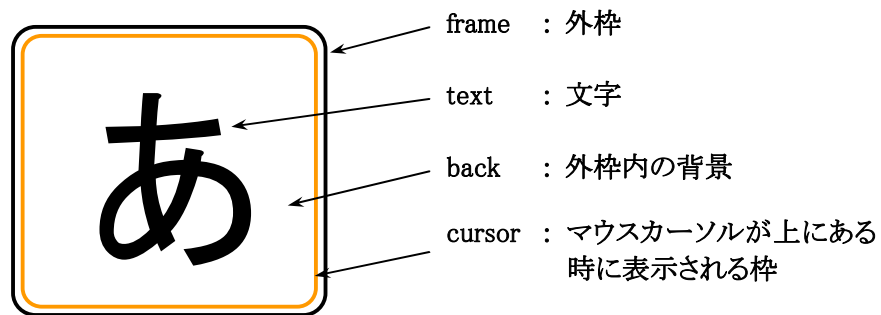


図 13 オブジェクトの各部の名称

※COLORREF の値が CLR_INVALID の場合は無効な値を表し、設定時は値の変更が行われません。また、読み出し時は未使用である事を示しており、設定を行っても無視されます。

※COLORREF 型および CLR_INVALID については、「10.2.8 Windows 側で定義されている型」をご覧ください。

10.3. 関数詳細

10.3.1. KTC_Create

簡タッチのインスタンスを生成します。

関数

```
HANDLE _stdcall KTC_Create (void);
```

引数

ありません

戻り値

関数が成功すると、インスタンスのハンドルが返ります。

関数が失敗すると、NULL(0)が返ります。

解説

必ず最初に実行する関数です。この関数で戻り値は他の関数全てで使用されます。また、インスタンスを削除する KTC_Delete 関数を実行するまで保存をしなくてはなりません。

本関数が成功した場合、簡タッチの終了時には KTC_Delete 関数を実行してください。

10.3.2. KTC_Delete

簡タッチのインスタンスを削除します。

関数

```
KTC_ERR _stdcall KTC_Delete (HANDLE hKTC);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

本関数は簡タッチの使用が終了した場合に必ず実行しなければなりません。関数を実行した後は hKTC に指定したハンドルは無効となり使用できなくなります。

※イベント通知の処理ハンドラの中で本関数を呼び出してインスタンス削除はしないで下さい。

10.3.3. KTC_GenWindow, KTC_GenWindow2

簡タッチのウィンドウを生成します。

関数

```
KTC_ERR _stdcall KTC_GenWindow (HANDLE hKTC);  
KTC_ERR _stdcall KTC_GenWindow2 (HANDLE hKTC, HWND hWndParent);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

hWndParent

親ウィンドウのハンドルを指定します。

NULL を指定するとデスクトップウィンドウが親ウィンドウになります。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

簡タッチのウィンドウが生成されます。本関数の実行直後のウィンドウは非表示状態です。表示状態を切り替えるには KTC_Show 関数を使用します。

KTC_GenWindow を使用した場合、メインアプリケーションウィンドウが親になります。メインアプリケーションウィンドウは通常、アプリケーションが最初に生成するウィンドウです。

この場合、たとえばメインアプリケーションウィンドウが非表示になると、簡タッチウィンドウも非表示になります。

KTC_GenWindow2 で簡タッチウィンドウを生成した場合、親ウィンドウをメインアプリケーション以外に変更できます。hWndParent を NULL にして実行した場合、親ウィンドウはデスクトップウィンドウになり、メインアプリケーションウィンドウが非表示になっても簡タッチウィンドウは非表示なりません。

既に簡タッチウィンドウを生成しているインスタンスにこの関数を実行すると、実行エラー (KTCERR_FAILED) となります。

10.3.4. KTC_Show

ウィンドウの表示状態を変更します。

関数

```
KTC_ERR _stdcall KTC_Show (HANDLE hKTC, KTC_ENSHOW cmdShow);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

cmdShow

ウィンドウの表示状態を指定します。(解説参照)

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

簡タッチウィンドウの表示状態を cmdShow で指定する状態に変更します。cmdShow には以下の値を指定できます。

表 22 KTC_ENSHOW 型で定義された名前

定義されている名前	意味
KTCSW_NORMAL	ウィンドウをアクティブにして表示します。ウィンドウが最小化または最大化されていた場合は、その位置とサイズを元に戻します。
KTCSW_MINIMIZE	ウィンドウを最小化します。
KTCSW_MAXIMIZE	ウィンドウを最大化します。
KTCSW_HIDE	ウィンドウを非表示にします。

最大化状態でウィンドウサイズを変更した場合(フレームやタイトルバーを変えた場合)は、一度 KTCSW_NORMAL を実行した後、再度 KTCSW_MAXIMIZE を実行する必要があります。タイトルバーが無い状態で最大化するとウィンドウの位置がずれることがあります。その場合は、KTC_SetDlgPosition で位置修正をして下さい。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー(KTCERR_FAILED)となります。

10.3.5. KTC_SetText、KTC_SetText2

文字表示枠に文字列を設定します。

関数

```
KTC_ERR _stdcall KTC_SetText (HANDLE hKTC, LPCSTR pBuf, long nBufLen);
```

```
KTC_ERR _stdcall KTC_SetText2 (HANDLE hKTC, LPCSTR pBuf, long nPos);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

pBuf

文字表枠の設定に使われる、NULL で終わる文字列へのポインタを指定します。

NULL の場合、文字表示枠の文字列を消去します。

nBufLen

バッファのバイト長を指定します。または、文字列のバイト長(終端の NULL は含まない)を指定します。0の場合、NULL 終端までの文字列を設定します。

nPos

カーソルの位置をバイト単位で指定します。指定した位置が全角文字の上である場合、カーソルはその文字の前になります。先頭位置は0で、負の値か文字列のバイト数以上の値を指定するとカーソルは最後尾に設定されます。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

既に文字表示枠に文字が有る場合は pBuf の文字列で置き換わります。nBufLen が pBuf の文字列長より小さい場合は nBufLen の長さで切られます。逆に nBufLen が大きい場合は pBuf の文字列は切られずに設定されます。

(注意) 文字表示枠に設定される文字列は、KTC_SetAction 関数の KTCA_LIMITTEXT を使用して設定したバイト数で切られます。

「7.12 文字表示枠への文字列設定」もご覧ください。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.6. KTC_GetText

文字表示枠の文字列を取得します。

関数

```
long _stdcall KTC_GetText (HANDLE hKTC, LPSTR pBuf, long nBufLen);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

pBuf

バッファへのポインタを指定します。このバッファに文字表示枠の文字列が格納されます。

nBufLen

バッファのバイト長を指定します。

戻り値

関数が成功すると、バッファに格納した文字列のバイト長(終端の NULL を含まず)が返ります。関数が失敗すると、KTC_ERR 型で定義されている KTCERR_OK 以外のエラーが返ります。

解説

文字表示枠の文字列を読み出して pBuf の示すバッファに格納します。文字列の長さが nBufLen-1 を超える場合は nBufLen-1 の長さに切られ、最後に NULL を加えた文字列を格納します。

「7.12 文字表示枠への文字列設定」もご覧下さい。

本関数を KTC_GenWindow が実行される前に使用すると、戻り値として 0 が返ります。

10.3.7. KTC_SetNotify

通知機能の設定を行います。

関数

```
KTC_ERR _stdcall KTC_SetNotify (HANDLE hKTC, HWND hWnd, UINT nMsgNo,  
                                KTC_CALLBACK fCallback);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

hWnd

メッセージを受信するウィンドウのハンドルを指定します。メッセージを受信しない場合は NULL を指定します。

nMsgNo

簡タッチが送信すべきメッセージの番号を指定します。

fCallback

コールバック関数へのポインタを指定します。コールバックを使用しない場合は NULL を指定します。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

簡タッチの状況を通知するメッセージまたはコールバックを設定します。この関数が成功した時点から設定が有効となります。アプリケーションへの通知方法は、メッセージ方式またはコールバック方式のどちらかを選べます。メッセージ方式を使用する場合は、hWndとnMsgNoを設定してください。コールバック方式を使用する場合はfCallbackを設定してください。hWndとfCallbackのどちらもNULLの場合は、通知機能を停止します。また、どちらもNULLで無い場合は、メッセージ方式となります。

コールバック関数は以下の形式です。

```
long _stdcall fCallback ( UINT nMsgNo, KTC_ENREASON lReason, long lParam );
```

通知機能についての詳しい内容は、「8 イベント通知機能」を参照して下さい。

本関数は、ダイアログが非表示／表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.8. KTC_SetAction

様々な動作設定を行います。

関数

```
KTC_ERR _stdcall KTC_SetAction (HANDLE hKTC, KTC_ENACTION INum, long lData);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

INum

設定を行う動作項目を示す値を指定します。

lData

設定する値を指定します。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

INum には以下の値を指定できます。

表 23 KTC_ENACTION 型の定義

定義されている名前	意味	lData 設定値
KTCA_FRAME	ウィンドウ枠の種類や有無	0~3
KTCA_TITLEBAR	タイトルバーの有無や最大化・最小化ボタンの有効/無効	0~5
KTCA_EXTERNAL	外部/内部モード切り替え	0~2
KTCA_LISTBTN	リスト表示キーの有無	0, 1
KTCA_LIMITTEXT	文字表示枠に入力できる文字数の制限	0~1000
KTCA_MOVEWINDOW	クライアント領域でのマウスドラッグでウィンドウ移動の可否	0, 1
KTCA_IMEONOFF	IME ON/OFF キーの有無	0, 1
KTCA_CONVBTN	変換キーの有無	0, 1

lData に設定する値などの詳細については、「7 カスタマイズ」をご覧ください。

本関数は、ダイアログが非表示/表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.9. KTC_GetAction

動作設定状態を取得します。

関数

```
KTC_ERR _stdcall KTC_GetAction (HANDLE hKTC, KTC_ENACTION lNum, long *pData);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

lNum

取得を行う動作項目を示す値を指定します。

pData

取得した値を格納する変数へのポインタを指定します。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

lNum には以下の値を指定できます。

表 24 KTC_ENACTION 型の定義

定義されている名前	意味	pData 取得値
KTCA_FRAME	ウィンドウ枠の種類や有無	0~3
KTCA_TITLEBAR	タイトルバーの有無や最大化・最小化ボタンの有効/無効	0~5
KTCA_EXTERNAL	外部/内部モード切り替え	0~2
KTCA_LISTBTN	リスト表示キーの有無	0, 1
KTCA_LIMITTEXT	文字表示枠に入力できる文字数の制限	1~1000
KTCA_MOVEWINDOW	クライアント領域でのマウスドラッグでウィンドウ移動の可否	0, 1
KTCA_IMEONOFF	IME ON/OFF キーの有無	0, 1
KTCA_CONVBTN	変換キーの有無	0, 1

pData に取得される値などの詳細については、「7 カスタマイズ」をご覧ください。

本関数は、ダイアログが非表示/表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.10. KTC_SetDlgPosition

ダイアログの位置・大きさを設定します。

関数

```
KTC_ERR _stdcall KTC_SetDlgPosition (HANDLE hKTC, KTC_POSITION *pPos);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

pPos

ダイアログの位置とサイズが格納された変数へのポインタを指定します。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

簡タッチのクライアントエリアの幅と高さの比率は常に一定です。そのため、幅と高さの内、より小さいウィンドウサイズになる方の値を使用します。また、幅もしくは高さに0を指定した場合は、0でない方の値を使用します。

本関数は、ダイアログが非表示／表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.11. KTC_GetDlgPosition

現在のダイアログの位置・大きさを取得します。

関数

```
KTC_ERR _stdcall KTC_GetDlgPosition (HANDLE hKTC, KTC_POSITION *pPos);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

pPos

取得した値を格納する変数へのポインタを指定します。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

本関数は、ダイアログが非表示／表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.12. KTC_SetDlgCaption

ダイアログのタイトルバーに表示されるキャプションを設定します。

関数

```
KTC_ERR _stdcall KTC_SetDlgCaption (HANDLE hKTC, LPCSTR *pCaption);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

pCaption

設定するキャプション文字列 (NULL で終わる文字列) へのポインタを指定します。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

「7.10 ウィンドウのキャプション」もご覧下さい。

本関数は、ダイアログが非表示／表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.13. KTC_SetDlgIcon

ダイアログのタイトルバーに表示されるアイコンを設定します。

関数

```
KTC_ERR _stdcall KTC_SetDlgIcon (HANDLE hKTC, HICON hIcon);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

hIcon

設定するアイコンのハンドルを指定します。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

hIcon に指定するハンドルは、Win32 の LoadIcon 関数や LoadImage 関数等で取得します。

hIcon に NULL を指定すると透明なアイコンが設定されます。(※透明ですがアイコンはありますので、アイコンをクリックした時に出るポップアップウィンドウは同様に出来ます)

「7.11 ウィンドウのアイコン」もご覧下さい。

本関数は、ダイアログが非表示／表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.14. KTC_SetList

リスト設定を行います。

関数

```
KTC_ERR _stdcall KTC_SetList (HANDLE hKTC, LPCSTR pList, long nListLen);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

pList

リストとして設定する文字列へのポインタを指定します。

nListLen

pList が指す文字列の長さを指定します。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

pList が指すリスト文字列は、リストの各行を NULL で終わる文字列として、1行目から連結したものです。リスト文字列の最後は NULL を 1 つ追加します。つまり、リスト文字列の最後は NULL が2つ続きます。

nListLen がリスト文字列より大きい場合は、リスト文字列全てが設定対象となります。また、nListLen がリスト文字列より小さい場合は、nListLen の長さで打ち切れ、最後の不完全な文字列(NULL で終わらない文字列)は設定されません。

設定可能なリストの最大行数は100行、各行の文字列長は最大100文字(最後のNULL文字を含まず)です。

「7.13 リスト設定」もご覧下さい。

本関数は、ダイアログが非表示／表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.15. KTC_UpdateView

画面を更新します。

関数

```
KTC_ERR _stdcall KTC_UpdateView (HANDLE hKTC);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

KTC_SetCaption 関数、KTC_SetFont 関数、KTC_SetColor 関数による変更は直ちに簡タッチ画面に反映されませんので、この関数により更新してください。

本関数は、ウィンドウが非表示／表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.16. KTC_SetKeyCode

キーの文字コードとIME モードを設定します。

関数

```
KTC_ERR _stdcall KTC_SetKeyCode (HANDLE hKTC, long nObjNo, long nTblNo, long nSubTblNo,  
                                LPCSTR pCode, long nAttr);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

nObjNo

キーのオブジェクト番号を指定します。

nTblNo

メインテーブル番号を指定します。

nSubTblNo

サブテーブル番号を指定します。

pCode

文字コードが格納されたバッファへのポインタを指定します。文字コードの終わりは NULL(0)です。

nAttr

文字コードを送信する際に設定する IME のモードを指定します。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

nObjNo、nTblNo、nSubTblNo で指定されるキーの文字コードとIME モードを設定します。

詳しくは「7.17 文字コード」をご覧ください。

本関数は、ウィンドウが非表示／表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.17. KTC_GetKeyCode

キーの文字コードとIME モードを取得します。

関数

```
KTC_ERR _stdcall KTC_GetKeyCode (HANDLE hKTC, long nObjNo, long nTblNo, long nSubTblNo,  
                                LPSTR pCode, long lLen, long *pnAttr);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

nObjNo

キーのオブジェクト番号を指定します。

nTblNo

メインテーブル番号を指定します。

nSubTblNo

サブテーブル番号を指定します。

pCode

文字コードを格納するバッファへのポインタを指定します。文字コードの終わりは NULL(0)です。

lLen

pCode が指すバッファの大きさをバイト数で指定します。

pnAttr

IME のモードを格納する変数へのポインタを指定します。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

nObjNo、nTblNo、nSubTblNo で指定されるキーの文字コードとIME モードを取得します。

詳しくは「7.17 文字コード」をご覧ください。

本関数は、ウィンドウが非表示／表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.18. KTC_SetCaption

キーのキャプションを設定します。

関数

```
KTC_ERR _stdcall KTC_SetCaption (HANDLE hKTC, long nObjNo, long nTblNo, long nSubTblNo,  
                                LPCSTR pCaption);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

nObjNo

キーのオブジェクト番号を指定します。

nTblNo

メインテーブル番号を指定します。

nSubTblNo

サブテーブル番号を指定します。

pCaption

キャプション文字列が格納されているバッファへのポインタを指定します。文字列の終わりは NULL(0)です。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

nObjNo、nTblNo、nSubTblNo で指定されるキーのキャプションを設定します。キャプションのフォントは KTC_SetFont 関数で設定します。

詳しくは「7.14 キャプション」をご覧ください。

本関数は、ウィンドウが非表示／表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.19. KTC_GetCaption

キーのキャプションを取得します。

関数

```
KTC_ERR _stdcall KTC_GetCaption (HANDLE hKTC, long nObjNo, long nTblNo, long nSubTblNo,  
                                LPSTR pCaption, long lLen);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

nObjNo

キーのオブジェクト番号を指定します。

nTblNo

メインテーブル番号を指定します。

nSubTblNo

サブテーブル番号を指定します。

pCaption

キャプション文字列を格納するバッファへのポインタを指定します。文字列の終わりは NULL(0)です。

lLen

pCaption が指すバッファの大きさをバイト数で指定します。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

nObjNo、nTblNo、nSubTblNo で指定されるキーのキャプションを取得します。キャプションのフォントは KTC_GetFont 関数で取得します。

詳しくは「7.14 キャプション」をご覧ください。

本関数は、ウィンドウが非表示／表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.20. KTC_SetFont

キーのキャプションのフォントを設定します。

関数

```
KTC_ERR _stdcall KTC_SetFont (HANDLE hKTC, long nObjNo, long nTblNo, long nSubTblNo,  
                             KTC_FONT *pFont);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

nObjNo

キーのオブジェクト番号を指定します。

nTblNo

メインテーブル番号を指定します。

nSubTblNo

サブテーブル番号を指定します。

pFont

フォント情報が格納された変数へのポインタを指定します。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

nObjNo、nTblNo、nSubTblNo で指定されるキーのキャプションのフォントを設定します。

詳しくは「7.15 フォント」をご覧ください。

本関数は、ウィンドウが非表示／表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.21. KTC_GetFont

キーのキャプションのフォントを取得します。

関数

```
KTC_ERR _stdcall KTC_GetFont (HANDLE hKTC, long nObjNo, long nTblNo, long nSubTblNo,  
                             KTC_FONT *pFont);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

nObjNo

キーのオブジェクト番号を指定します。

nTblNo

メインテーブル番号を指定します。

nSubTblNo

サブテーブル番号を指定します。

pFont

フォント情報を格納する変数へのポインタを指定します。構造体メンバの len には、同メンバの pFaceName が指すバッファの大きさを指定しなければなりません。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

nObjNo、nTblNo、nSubTblNo で指定されるキーのキャプションのフォントを取得します。

詳しくは「7.15 フォント」をご覧ください。

本関数は、ウィンドウが非表示／表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.22. KTC_SetColor

キーと画面全体の色を設定します。

関数

```
KTC_ERR _stdcall KTC_SetColor (HANDLE hKTC, long nObjNo, KTC_COLOR *pColor0,  
                               KTC_COLOR *pColor1, KTC_COLOR *pColor2);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

nObjNo

対象のオブジェクト番号を指定します。

pColor0、pColor1、pColor2

色情報が格納された変数へのポインタを指定します。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

nObjNo で指定される対象のオブジェクトに対して、色を設定します。pColor0～2 は、対象とするオブジェクトの動作状態によって使い分けられます。たとえば、テーブル選択キーの場合、pColor2は選択されている時の色、pColor1 はキャプションが空の時の色、pColor0 はその他の色(通常の色)として使用されます。

詳しくは「7.16 色」をご覧ください。

本関数は、ウィンドウが非表示／表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.23. KTC_GetColor

キーと画面全体の色を取得します。

関数

```
KTC_ERR _stdcall KTC_SetColor (HANDLE hKTC, long nObjNo, KTC_COLOR *pColor0,  
                               KTC_COLOR *pColor1, KTC_COLOR *pColor2);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

nObjNo

対象のオブジェクト番号を指定します。

pColor0、pColor1、pColor2

色情報を格納する変数へのポインタを指定します。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

nObjNo で指定される対象のオブジェクトに対して、色を取得します。

詳しくは「7.16 色」をご覧ください。

本関数は、ウィンドウが非表示／表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.24. KTC_Save

簡タッチの設定状態を記録します。

関数

```
KTC_ERR _stdcall KTC_Save (HANDLE hKTC, LPCSTR pFileName, long lSave);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

pFileName

記録先のファイル名をフルパス名で指定します。ファイル名の終わりは NULL(0)です。

lSave

記録範囲を示す値を指定します。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

記録範囲は C ヘッダーファイル上に KTCSV_xxx として定義されており、それぞれを論理ORすることで複数範囲を指定することができます。詳しくは「9 状態の記録と復帰」をご覧ください。

本関数は、ウィンドウが非表示／表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.25. KTC_Restore

簡タッチの設定状態を復帰します。

関数

```
KTC_ERR _stdcall KTC_Restore (HANDLE hKTC, LPCSTR pFileName, long lSave);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

pFileName

記録ファイル名をフルパス名で指定します。ファイル名の終わりは NULL(0)です。

lSave

復帰範囲を示す値を指定します。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

復帰範囲は C ヘッダーファイル上に KTCSV_xxx として定義されており、それぞれを論理ORすることで複数範囲を指定することができます。詳しくは「9 状態の記録と復帰」をご覧ください。

本関数は、ウィンドウが非表示／表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.26. KTC_GetWindowHandle

簡タッチウィンドウのハンドルを取得します。

関数

```
KTC_ERR _stdcall KTC_GetWindowHandle (HANDLE hKTC, HANDLE* phCDialog, HWND* phWnd);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

phCDialog

CDialog 型ハンドルの格納先ポインタ。NULL を指定すると取得されません。

phWnd

HWND 型ハンドルの格納先ポインタ。NULL を指定すると取得されません。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

本関数では簡タッチウィンドウへのハンドルとして2種類の値が取得できます。アプリケーション開発において、MFC を利用できる場合は CDialog 型ポインタを利用すると便利です。CDialog 型ポインタは、HANDLE 型として取得されますので、CDialog 型ポインタにキャストして使用して下さい。MFC を利用できない場合は HWND 型のポインタを利用します。

本関数は、ウィンドウが非表示／表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.27. KTC_SetTableNum

テーブル番号を設定します。

関数

```
KTC_ERR _stdcall KTC_SetTableNum (HANDLE hKTC, long nTblNo, long nSubTblNo);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

nTblNo

メインテーブル番号を指定します。

nSubTblNo

サブテーブル番号を指定します。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

テーブル番号を設定し、簡タッチ画面の表示更新を行います。

本関数は、ウィンドウが非表示／表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

10.3.28. KTC_GetTableNum

テーブル番号を取得します。

関数

```
KTC_ERR _stdcall KTC_GetTableNum (HANDLE hKTC, long *pnTblNo, long *pnSubTblNo);
```

引数

hKTC

簡タッチのインスタンスのハンドルを指定します。このハンドルは KTC_Create 関数で取得します。

pnTblNo

メインテーブル番号を格納する変数へのポインタを指定します。

pnSubTblNo

サブテーブル番号を格納する変数へのポインタを指定します。

戻り値

関数が成功すると、KTCERR_OK が返ります。

関数が失敗すると、KTCERR_OK 以外が返ります。

解説

現在設定されているテーブル番号を取得します。

本関数は、ウィンドウが非表示／表示状態に関わらず実行可能です。

本関数を KTC_GenWindow が実行される前に使用すると、実行エラー (KTCERR_FAILED) となります。

11. Appendix

11.1. コード例

簡タッチを独立して動かすコードの例を示します。

開発環境は Visual C++[®] 6.0 の Win32 Application です。

この例では、閉じるボタンが押されると終了するように、コールバックルーチンを設定しています。

◇ .CPP ファイル

```
#include "stdafx.h"
#include "Ktouch.h"

//*****
// グローバル変数
//
HANDLE g_hKTC = NULL; // 簡タッチへのハンドル

//*****
// CKTCBase のイベントコールバックルーチン
//
long _stdcall KtcCallback( UINT nMsgNo, KTC_ENREASON IReason, long IParam )
{
    switch( IReason )
    {
        case KTCR_BTNCLOSE:
            PostQuitMessage( 0 );
            return 0;
    }
    return 0;
}

//*****
// メインルーチン WinMain
//
int APIENTRY WinMain(HINSTANCE hInstance,
                    HINSTANCE hPrevInstance,
                    LPSTR lpCmdLine,
                    int nCmdShow )
{
    MSG msg;
    //-----
    // 簡タッチの準備
    //
    g_hKTC = KTC_Create();
    if( g_hKTC == NULL ) return FALSE;
    KTC_GenWindow( g_hKTC );
    //-----
    // 簡タッチのウィンドウの大きさ設定
    //
    KTC_POSITION ktcp;
    KTC_GetDlgPosition( g_hKTC, &ktcp );
    ktcp.Width = 480;
    ktcp.Height = 0;
}
```

```
KTC_SetDlgPosition( g_hKTC, &ktcpos );
//-----
// 簡タッチの外観設定
//
KTC_SetAction( g_hKTC, KTCA_TITLEBAR, 4);
//-----
// 簡タッチの通知メッセージ設定と表示開始
//
KTC_SetNotify( g_hKTC, NULL, 0, KtcCallback );
KTC_Show( g_hKTC, KTCSW_NORMAL );
//-----
// メイン メッセージ ループ
//
while( GetMessage(&msg, NULL, 0, 0) )
{
    TranslateMessage( &msg );
    DispatchMessage( &msg );
}
//-----
// メイン メッセージ終了時の処理
//
KTC_Delete( g_hKTC );
return msg.wParam;
}
```

12. 既知の不具合

現在、既知になっている不具合について記載致します。

12.1. IME 制御の不具合

ご利用のアプリケーションや OS のバージョン等により IME の制御が行えないという不具合が発生しています。現在判明しているアプリケーションは Microsoft Word および WordPad です。

この場合、簡タッチでは以下のような不具合が発生します。

- ① 簡タッチで文字を入力したが、変換入力になるはずが確定入力になってしまう。
- ② 変換入力状態が切り替わっても、イベント通知 (KTCCR_IMESTATUS) がアプリに来ない。
- ③ 変換入力状態の ON/OFF により、スペースキーと変換キーのキャプションが切り替わる機能が働かない。

①の対策として、V1.5 からは IME ON/OFF キーと変換キーを設けました。これらのキーは KTC_SetAction により表示／非表示の制御ができます。

②と③の対策としては残念ながら有効な手段がありません。不具合が発生するアプリケーションをご利用になる場合には、当該の機能を使用しないようにお願いします。

12.2. .NET 版 Visual Basic ご利用時の不具合

.Net 版の Visual Basic から内部モードの簡タッチを使用すると、文字表示枠に入力される文字が化けることがあります。

この不具合については簡タッチの修正内容が大きいため、本バージョンアップにおいては対策を施していません。次回バージョンアップで対策予定です。

13. サポート情報

製品に関する情報、最新のバージョンなどは弊社ホームページにてご案内しております。また、お問い合わせ、ご質問などは下記までご連絡ください。(弊社ホームページに「お問い合わせフォーム」があります)

テクノウェーブ(株)

URL : <http://www.techw.co.jp>

E-mail : support@techw.co.jp

改訂記録

年月	版	改訂内容
2005年8月	初	Ver1.0として新規作成
2007年4月	2	Ver1.4用に更新
2008年4月	3	Ver1.5用に更新