

DT-950 シリーズ

コモンコントロール開発ガイド

概要説明

このガイドは、DT-950 のアプリケーション開発者向けの開発ガイドブックです。DT-950 ではタッチパネルがなく、画面サイズが小さいため、Windows 標準のコントロールの代わりとしてコモンコントロールを提供します。本書では、コモンコントロールの使い方について説明します。



ご注意

このソフトウェアおよびマニュアルの一部または全部を無断で使用、複製することはできません。このソフトウェアおよびマニュアルは、本製品の使用許諾契約書のもとでのみ使用することができます。

このソフトウェアおよびマニュアルを運用した結果の影響については、一切の責任を負いかねますのでご了承ください。

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

このマニュアルの著作権はカシオ計算機株式会社に帰属します。

本書中に含まれている画面表示は、実際の画面とは若干異なる場合があります。予めご了承ください。

© 2005 カシオ計算機株式会社

Microsoft, MS, ActiveSync, Active Desktop, Outlook, Windows, Windows NT, および Windows ロゴは、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。Microsoft 社の製品は、OEM 各社に、Microsoft Corporation の 100%出資子会社である Microsoft Licensing, Inc.によりライセンス供与されています。

目次

1. DT-950 コモンコントロールの種類	1-1
2. サンプルアプリケーション	2-1
2-1.動作	2-1
2-2.作成上の注意点	2-7
3. コモンコントロール使用方法	3-1
4. 関数リファレンス	4-1

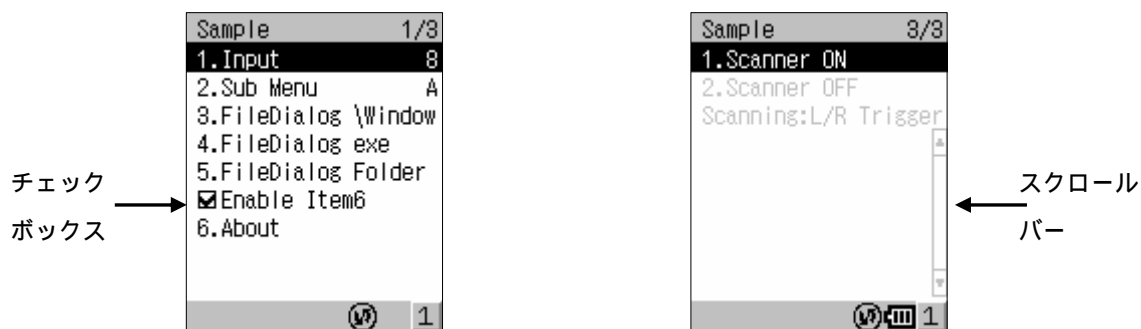
1. DT-950 コモンコントロールの種類

本機には以下のコモンコントロールがあります。

メニューコントロール/ベースウィンドウ/メッセージボックス/エディットコントロール/ファイル選択コントロール/砂時計カーソル

以下は各コントロールの例です。

- ・メニューコントロール

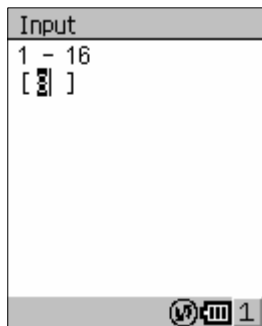


アプリケーションのメニューを表示する画面です。

チェックボックス、スクロールバーはメニューコントロールのパラメーターで表示します。

スクロールバーは上下キーで動かします。

- ・ベースウィンドウ/エディットコントロール



左図の[]内の数値入力部分がエディットコントロールです。

エディットコントロールが張り付いている画面がベースウィンドウです。タイトルバーの Input、メイン画面の 1-16 と入力範囲を表す[]はベースウィンドウに表示しています。

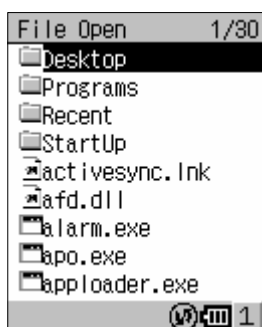
ベースウィンドウはメニュー以外の画面作成に使用します。

- ・メッセージボックス



メッセージを表示する画面です。

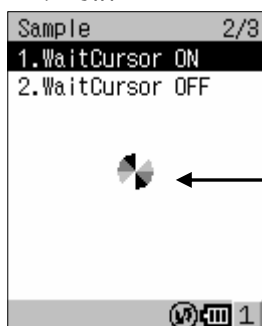
・ファイル選択コントロール



ファイルエクスプローラーの様にフォルダ、ファイルを表示します。

実行時のモードを指定でき、タイトルバーに現在のモードとページ番号が表示されます。左図はオープンモードです。

・砂時計カーソル



左図の真中が砂時計カーソルです。
処理実行中を表すのに使います。

砂時計
カーソル

2. サンプルアプリケーション

コモンコントロールを使ったサンプルアプリケーション “ControlSample” プロジェクトを提供します。アプリケーションの動作と作成上の注意点を記します。

2-1. 動作

サンプルアプリケーションの動作についてアプリケーションの各ページの各項目ごとに説明します。

製品のデフォルトではマルチキーは Enter キーの機能が割りあっているため、以下で Enter キー入力部分はマルチキーでも同じです。

トップメニュー

アプリケーション起動時のトップメニュー（画面[0]）です。

使用コントロール：メニューコントロール

1/3 ページ

① Input

トップメニューで、1 を選択すると Input 画面（画面[1-1]）が表示されます。

1～16までの数を入力するエディットコントロールの例です。1～16の数を入力し（画面[1-2]）、決定するとトップメニューに戻ります（画面[1-3]）。トップメニューの1. Input の右側に入力した数2が表示されます。

エディットコントロールはベースウィンドウ上に貼り付けています。エディットコントロールは入力機能のみで、画面の他の表示はベースウィンドウで設定しています。

使用コントロール：ベースウィンドウ、エディットコントロール

① Sub menu

トップメニューで、2 を選択すると Sub Menu 画面（画面[2-1]）が表示されます。

トップメニューから選択した項目でメニュー（サブメニュー）を表示する例です。ItemA～ItemC を選択するとトップメニューに戻ります（画面[2-2]）。トップメニューの2. Sub Menu の右側に選んだアルファベット B が表示されます。

使用コントロール：メニューコントロール

② FileDialog ¥Window

トップメニューで、3 を選択すると Windows フォルダが開きます（画面[3-1]）。Windows フォルダ内のファイルとフォルダが全て表示されます。ファイルを開くモードで実行しています。（タイトル行の左上に File Open と表示）ファイルを1つ選択し（画面[3-2]）、決定するとトップメニューに戻ります（画面[3-3]）。トップメニューの最下行に、選択したファイル名（パスを含む）¥windows¥activesync. lnk が表示されます。ファイルを選択せずにトップメニューに戻った場合は7行目に FALSE が表示されます。

使用コントロール：ファイル選択コントロール

③ FileDialog exe

トップメニューで、4を選択するとルートフォルダが開きます（画面[4-1]）。ファイルの拡張子 exe のみに制限するモードで開いているため、exe ファイルとフォルダが表示されます。また、ファイルを保存するモードで実行しています（タイトル行の左上に File Save と表示）。「Fn」「R」キーを続けて入力し、FlashDisk フォルダを開きます（画面[4-2]）。FlashDisk のファイル ControlSample.exe を保存するファイルに決定すると（画面[4-3]）、保存するファイル名を入力するエディットコントロールが表示されます（画面[4-4]）。ファイル名を決定すると、トップメニューに戻り、最下行に入力されたファイル名が表示されます（画面[4-5]）。

このモードは保存するファイルとその名前の決定のみで保存処理はしていません。

使用コントロール：ファイル選択コントロール

④ FileDialog Folder

トップメニューで、5を選択するとルートフォルダが開きます（画面[5-1]）。フォルダ選択をするモードで実行しているため（タイトル行の左上に Folder Select と表示）、フォルダのみ表示されます。フォルダを選択し（画面[5-2]）、決定するとトップメニューに戻ります（画面[5-3]）。トップメニューの最下行に選択したフォルダ名 My Documents が表示されます。

使用コントロール：ファイル選択コントロール

⑤ Enable Item6

トップメニューの5、6の間にチェックボックスがあります。このチェックボックスは、トップメニューの「6. About」項目の有効／無効を設定します。Enter 入力でチェックを外すと（画面[7-1]）、「6. About」項目が無効になりグレー表示に変わります。

⑥ About

トップメニューで6を選択するとメッセージボックスが表示されます（画面[6-1]）。このメッセージボックスはプログラムのバージョンを確認するもので、ENT 入力でトップメニューに戻ります（画面[6-2]）。

使用コントロール：メッセージボックス

2/3 ページ

⑦ WaitCursor ON

以下の動作で2/3ページのメニューが表示されます（画面[8-1]）。

- ・ トップメニューでRを選択する
- ・ 1ページの最後の項目以降も上下キーの下キーを入力する。

1を選択すると、砂時計カーソルが回転し始めます。（画面[8-2]）

使用コントロール：砂時計カーソル

⑧ WaitCursor OFF

砂時計カーソル回転時に、2/3ページのメニューで2を選択すると（画面[8-3]）、砂時計カーソルは非

表示になります（画面[8-4]）。

使用コントロール：砂時計カーソル

3/3 ページ

⑨ Scanner ON

トップメニューでRを2回選択（または1/3ページから下キーを続けて入力）すると、3/3ページのメニューが表示されます（画面[9-1]）。

1を選択すると、「2. Scanner OFF」がアクティブになり、「1. Scanner ON」はグレー表示になります（画面[9-2]）。

この画面では、レーザースキャナをオープン終了し、バーコードを読める状態になっています。（レーザースキャナのオープン失敗したときには、エラーになりメッセージボックスが表示されます。例えばIrDA使用時のオープンなどです）

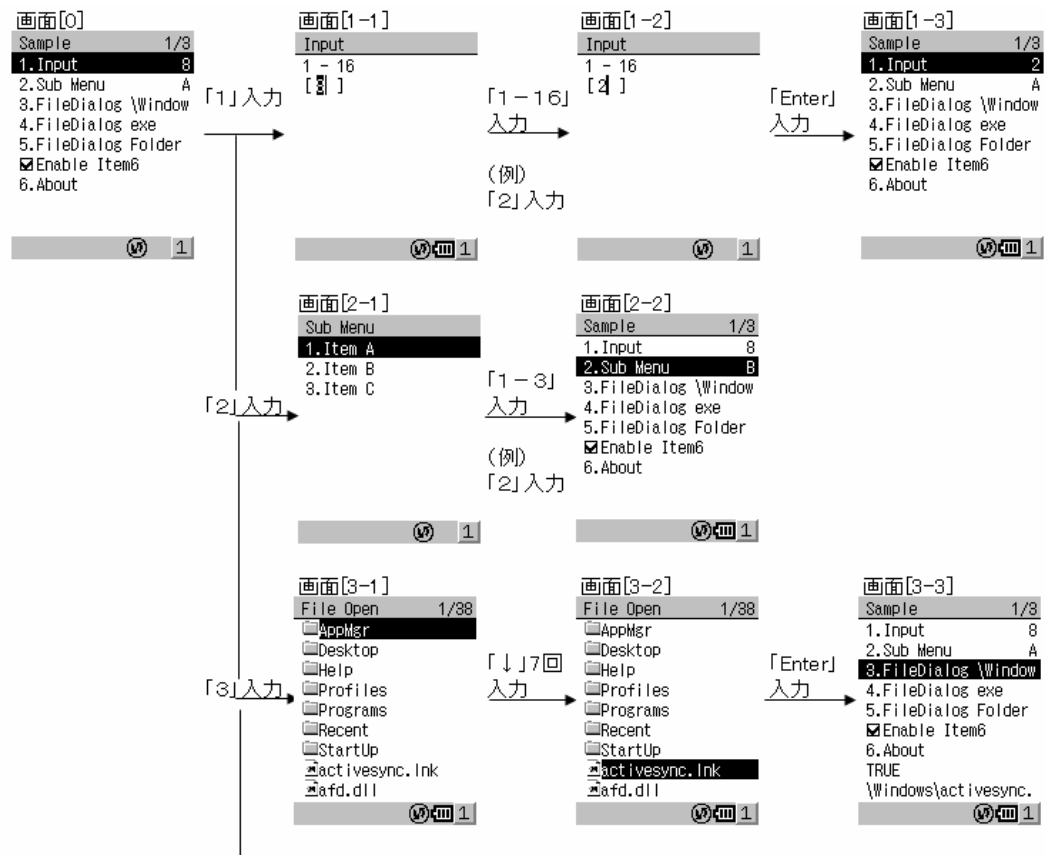
バーコードを読むと、メニュー画面のスクロールバーの行以降に表示されます。（画面[9-3]）

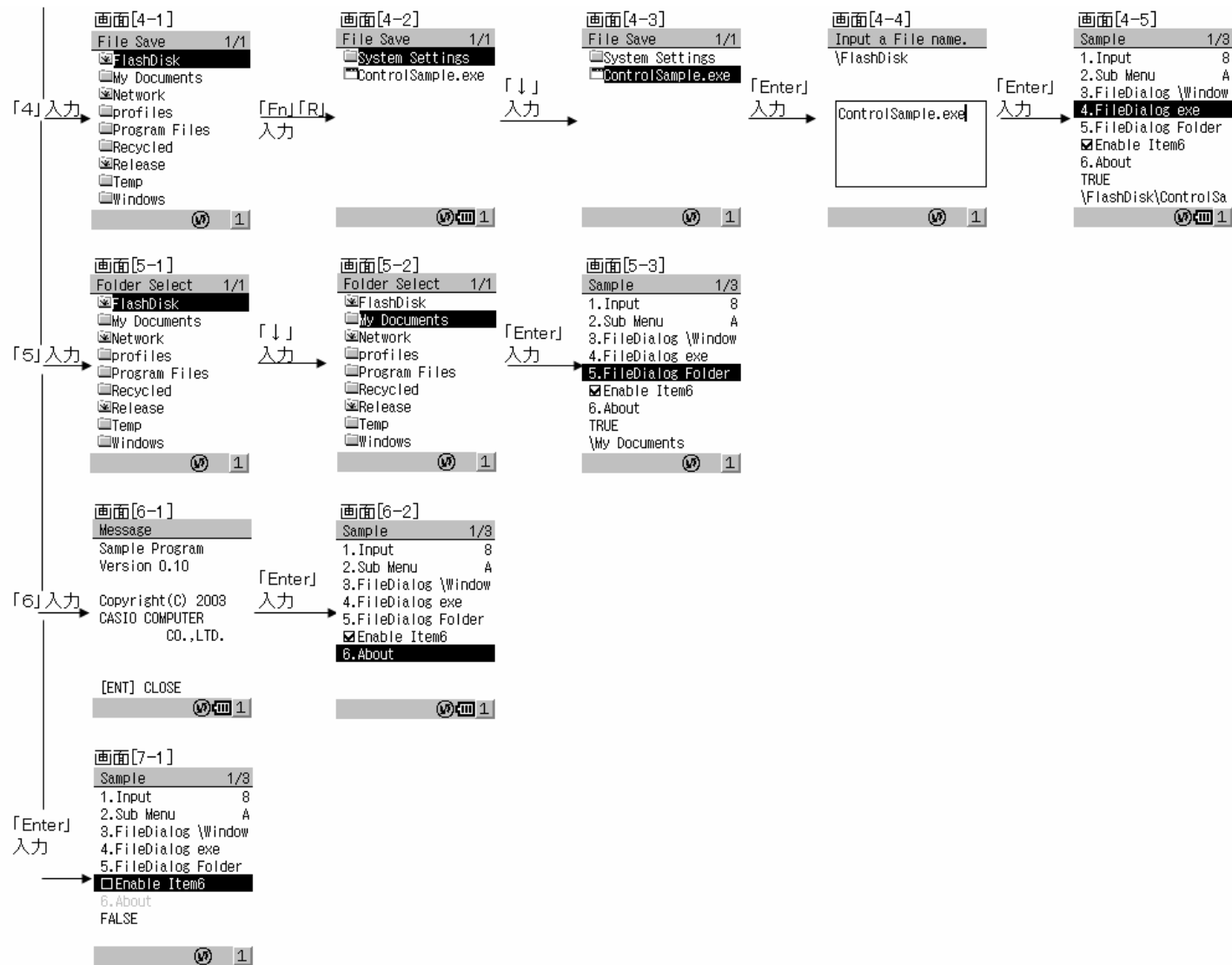
⑩ Scanner OFF

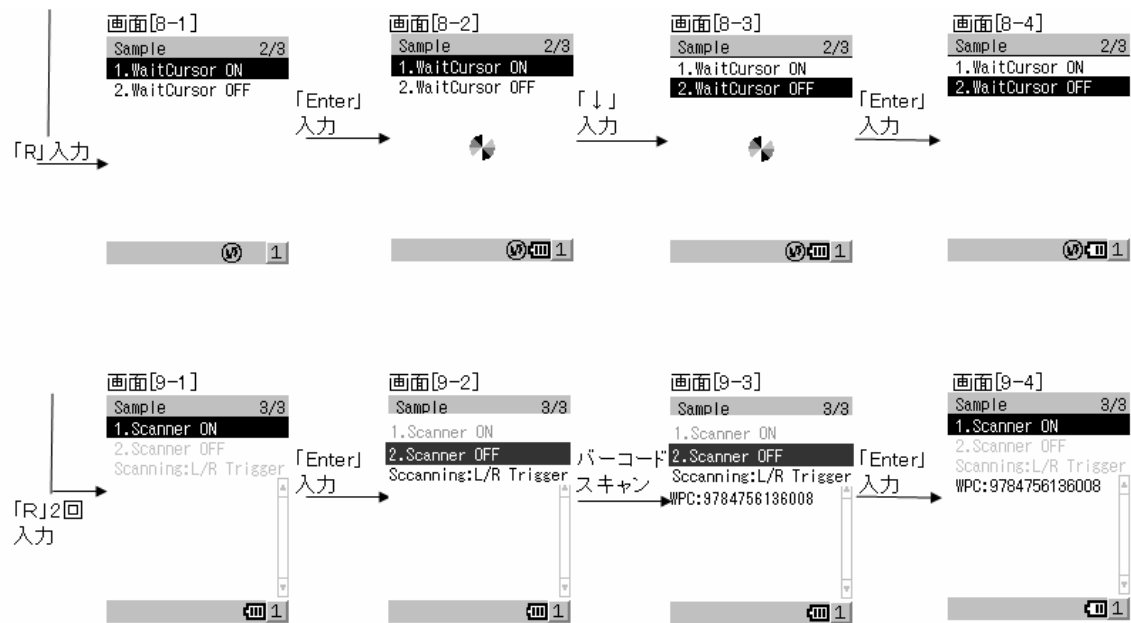
スキャナオン状態で3/3ページのメニューで2を選択すると（画面[9-3]）、「1. Scanner ON」がアクティブになり、「2. Scanner OFF」はグレー表示になります（画面[9-4]）。

この画面では、レーザースキャナをクローズしバーコードを読めない状態になっています。

以下はアプリケーションの画面遷移です。







Enter 入力はマルチキーでも同じ動作です。(マルチキーがデフォルト設定の場合です)

2-2. 作成上の注意点

コモンコントロールを使うのに、注意すべき部分を説明します。

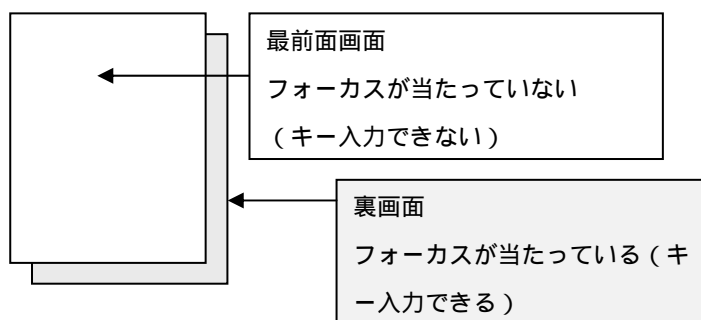
以下の注意点番号とサンプルソース ControlSample.cpp のコメントに同じ番号を付けています。

入力フォーカスについて

本機で最前面のウィンドウが入力フォーカスを失い最前面より裏のウィンドウが入力フォーカスを持ってしまうと、キー入力が裏のウィンドウに入るため動作できなくなってしまいます。(下図) 入力デバイスにタッチパネルがあればウィンドウを触りフォーカスを変えられますが、本機にはタッチパネルがないため入力フォーカスに注意してアプリケーションを作成する必要があります。

そのため、アプリケーションで注意点 1 のフォーカス処理とをしています。新規アプリケーション作成時と同じ処理を入れてください。

(フォーカスを失った状況)



最前面のウィンドウとフォーカスを持ったウィンドウがずれてしまっている状態

[注意点 1]

この処理は最前面アプリケーションから入力フォーカスが外れた後に、再度アプリケーションに来たフォーカスをアプリケーションの適切なウィンドウに当てるためのものです。以下の様な動作がこれにあたります。

(例)

- ・アプリケーション動作中にシステムから警告などのメッセージが来て、メッセージを終了した後
- ・アプリケーション動作中にシステムメニューを表示し (Fn、F4 を連続で入力)、再度アプリケーションに戻ったとき

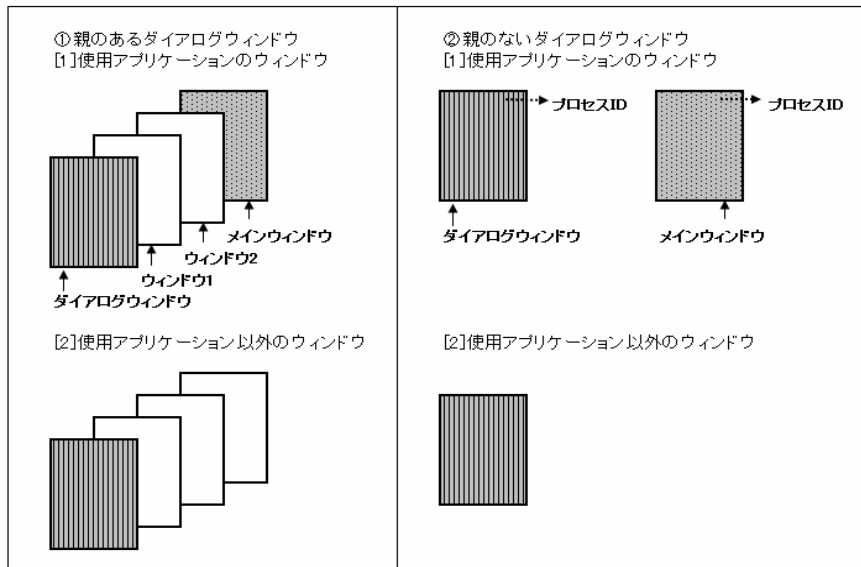
・適切なウィンドウの決め方

ダイアログウィンドウ (メッセージボックス、ファイル選択コントロール、Windows 標準ダイアログボックス) が表示されている場合には、ダイアログウィンドウがフォーカスを持つべきウィンドウになります。ダイアログウィンドウをを閉じないと他の処理に移らないからです。

ダイアログウィンドウがない場合にはメニューウィンドウがフォーカスを持つべきウィンドウになります。

以下にダイアログウィンドウの有無の判定方法とメニューウィンドウの決め方について示します。

・ダイアログウィンドウの有無の判定



ダイアログウィンドウは EnumWindows 関数で検索します。この関数は使用アプリケーションのダイアログウィンドウがあった場合にはダイアログウィンドウのウィンドウハンドルを返します。

EnumWindows 関数は使用アプリケーションのみではなく、全てのウィンドウ（子ウィンドウは除く）に対して検索します。アプリケーションで表示されている場合に前面に出すダイアログウィンドウは使用アプリケーションのウィンドウ [1]と [1]の場合です。

[1]の検索

上図の様にダイアログウィンドウの親ウィンドウをたどっていくとアプリケーションの初めで作成したメインウィンドウがあり、ダイアログウィンドウとメインウィンドウに関係（孫、ひ孫など）があることがわかります。

[2]のように使用アプリケーション以外のダイアログウィンドウは、親ウィンドウをたどっていても使用アプリケーションのメインウィンドウと関係がありません。

上の関係を利用して、検索対象のウィンドウの親ウィンドウをたどり親ウィンドウのハンドルがアプリケーション作成時のメインウィンドウのハンドルと一致した場合には、使用中のアプリケーション表示のダイアログウィンドウがあると判別します。

サンプルアプリケーションでは、検索対象のウィンドウから 10 世代分の親ウィンドウまでをたどっています。

[1]の検索

親ウィンドウを指定しないメッセージボックスなど親ウィンドウのないダイアログウィンドウもあります。親ウィンドウがないもので、使用中アプリケーション所有かどうかを判別するにはプロセスIDを使用します。検索対象のウィンドウのプロセスIDとメインウィンドウのプロセスIDを比較し、一致した場合は使用中のアプリケーションのウィンドウと判断します。

・メニューウィンドウの決め方

使用中のアプリケーションでダイアログウィンドウが未表示だった場合には、最前面に表示していたメニューウィンドウにフォーカスを当てます。そのために、アプリケーションでメニュー現状のメニューのウィンドウハンドルを保存しておきます。

(ControlSample.cpp の[注意点])

[注意点]

メッセージボックス/ファイル検索ダイアログ/ダイアログボックス (Windows 標準) 表示し終了した後は、最前面の画面にフォーカスが当たらなくなります。そのため、関数後はフォーカスを SetFocus 関数で最前面の画面に当ててください。

[注意点]

ウィンドウを切り替えるときには、現在のウィンドウを非表示にして、新しいウィンドウを表示してください。

ShowWindow(非表示にするウィンドウハンドル, SW_HIDE)

ShowWindow(表示にするウィンドウハンドル, SW_SHOW)

これを入れないとフォーカスのみ変更され、画面が表示されずフォーカスを失った状態になるので注意してください。

アプリケーションがフォーカスを失ってしまった場合

アプリケーション作成中などにフォーカスを失ってしまった場合には以下の方法でアプリケーションを終了させてください。

Fn と F4 を続けて入力してシステムメニューを表示してください。

CRL キーを入力し、Top Menu を表示し、3.Task Manager を選択してください。

Task Manager の画面で実行中のアプリケーション名が表示されているので選択してください。

2.End Task を選択してください。(アプリケーションが終了します)

3. コモンコントロール使用方法

動作環境

機種 DT-950

OS Microsoft WindowsCE .NET 4.1

開発環境

Microsoft eMbedded C++ Version4.0 + SP1

Microsoft Visual Studio .NET 2003

提供ファイル

CommonCTL.lib	ダイナミックライブラリ
CommonCTL.dll	インポートライブラリ
CTLDefine.h	ヘッダファイル

使用方法

アプリケーションは CTLDefine.h をインクルードし、CommonCTL.lib をリンクすることにより dll 内の関数を使用できます。(LoadLibrary() や GetProcAddress() 関数呼出は不要です。)

コモンコントロールが発行する独自メッセージ (UM_CTLXXXX) には、

WM_APP+0x3000 ~ WM_APP+0x30FF

を使用しています。

アプリケーションは番号が重ならないように注意してください。

4.関数リファレンス

以下は、コモンコントロールで使用する関数名と概要です。

機能	関数名	処理内容
メニュー	CTLCreateMenuControl	DT950 の基本画面であるメニュー表示画面を作成します
	CTLSetMenuParam	メニューコントロールのパラメータを再設定し、表示を更新します
	CTLGetMenuParam	メニューコントロールのパラメータを読み込みます
	CTLCloseMenuControl	メニューコントロールを破棄します
ベースウィンドウ	CTLCreateBaseWindow	メニューを表示しない画面を作成します
	CTLCloseBaseWindow	ベースウィンドウを破棄します
メッセージボックス	CTLMessageBox	メッセージを表示したい場合に使用します
エディットコントロール	CTLCreateEditControl	文字入力をさせたい場合に使用します
	CTLCloseEditControl	エディットコントロールを破棄します
ファイル選択コントロール	CTLFileDialogControl	ファイル選択を行う場合に使用します
メニュー (リソース)	CTLCreateMenuControlRC	機能は CTLCreateMenuControl と同じ
	CTLSetMenuParamRC	機能は CTLSetMenuParam と同じ
	CTLGetMenuRC	リソースで作成した情報を MENUWHOLE と MENUITEM 構造体に読み込みます
ベースウィンドウ (リソース)	CTLCreateBaseWindowRC	機能は CTLCreateBaseWindow と同じ
砂時計カーソル	CTLDispWaitCursor	擬似砂時計カーソルを表示します
	CTLUndispWaitCursor	擬似砂時計カーソルを破棄します

以下の 2 個の構造体はメニューコントロール／ベースウィンドウで使⽤します。

構造体構⽂

```
struct MENUWHOLE{ LPCTSTR lpTitle;  
                  int iDefCur;  
                  int iScroll;  
                  bool bPageN;  
                  bool bPageP;  
                  bool bArrowU;  
                  bool bArrowD;  
                  int IID;  
                  };
```

メニュー画面全体に関する情報を格納します。

メンバ

lpTitle : 表⽰したいタイトルを NULL で終わる文字列へのポインタを指定します。

単⾏のみ (全角 10 文字、半角 20 文字まで)

iDefCur : 初期表⽰でカーソルを当てる⾏を指定します。(1~9)

0 を指定した場合は 1 行⽬から数え bCursor、bActive が共に TRUE の⾏に設定されます。(すべての⾏が該当しない場合は CTL_NO_DEFAULT_LINE が指定されカーソルが表⽰されないページになります。) (CTLCreateBaseWindow 関数を呼び出す場合 : 設定不要)

iScroll : スクロールバーを伴うリスト部分の開始位置を指定します。(1~9)

スクロールバーを指定しない場合は 0 を指定する。(CTLCreateBaseWindow 関数を呼び出す場合 : 設定不要)

bPageN : 次ページの存在を指定します。(TRUE/FALSE)

(CTLCreateBaseWindow 関数を呼び出す場合 : 設定不要)

bPageP : 前ページの存在を指定します。(TRUE/FALSE)

(CTLCreateBaseWindow 関数を呼び出す場合 : 設定不要)

bArrowU : スクロールバーの上項の存在を指定します。(TRUE/FALSE)

(CTLCreateBaseWindow 関数を呼び出す場合 : 設定不要)

bArrowD : スクロールバーの下項の存在を指定します。(TRUE/FALSE)

(CTLCreateBaseWindow 関数を呼び出す場合 : 設定不要)

iID : メニューを識別するための ID 番号を指定します。

構造体構文

```
struct MENUITEM{ LPCTSTR lpLabel;  
                 int iKeyVal;  
                 bool bCursor;  
                 bool bActive;  
                 bool bCheck;  
                 };
```

メニュー画面の各行の情報を格納します。

メンバ

lpLabel : 表示したい項目(ラベル)を NULL で終わる文字列へのポインタを指定します。

単行のみ (全角 10 文字、半角 20 文字まで)

iKeyVal : 数字キーで選択する為に対応する数字キーを指定します。(CTL_KEY_00~CTL_KEY_09)

チェックボックス属性の場合は ON/OFF 状態を示します。(0:OFF、1:ON)

(CTLCreateBaseWindow 関数を呼び出す場合: 設定不要)

bCursor : TRUE:カーソル選択が有効に指定されます。

FALSE:カーソル選択が無効に指定されます。

(CTLCreateBaseWindow 関数を呼び出す場合: 設定不要)

bActive : TRUE:文字がアクティブに指定されます。

FALSE:文字が非アクティブ(薄色表示)に指定されます。

BCheck : TRUE:チェックボックス属性が指定されます。

FALSE:チェックボックスなしの通常状態が指定されます。

(CTLCreateBaseWindow 関数を呼び出す場合: 設定不要)

次頁より、関数リファレンスを説明します。

CTLCreateMenuControl

機能 DT950 の基本画面であるメニュー表示に使用します。
アプリから親ウインドウのハンドル、タイトル文字列、各行の文字列、項目毎のキーコードなどを指定して Create します。不要になったら Close します。リストビュー（詳細表示）のようなものです。

メニューの項目選択は、上下キーで移動する場合と番号入力する場合があります。
上下キー入力ではコントロールがカーソルを動かしますが、番号入力はコントロールでカーソルを動かしません。番号入力の場合はアプリでカーソルを移動してください。（参照：ControlSmample.cpp の SetMenuCur 関数）

項目が選択された時や CLR キーなどが押された時に、親ウインドウにメッセージを送信します。

カーソルが移動しない行や、項目をグレー表示する行などを設定することができます。
特殊な使い方としては、メニュー表示の一部分だけがスクロールするような画面が作成できます。

書式 `HWND CTLCreateMenuControl(HWND hwnd,
MENUWHOLE *pmw,
MENUITEM *pmi
);`

パラメータ

hwnd : Create するメニューの親ウインドウのハンドルを指定します。
(このウインドウにメニュー項目選択時にメッセージを送信します)

pmw : MENUWHOLE 構造体のポインタを指定します。

pmi : MENUITEM 構造体のポインタを指定します。

戻り値

関数が成功すると、作成されたウインドウのハンドルが返ります。

関数が失敗すると、NULL が返ります。

CTLCloseMenuControl

機能 メニューコントロールを破棄します。

書式 **bool CTLCloseMenuControl(HWND hwnd);**

パラメータ

hwnd : 破棄するメニューウインドウのハンドルを指定します。

戻り値

TRUE	正常終了
FALSE	エラー

メニューコントロールで利用されるメッセージ

UM_CTLMENU

メニューコントロールの項目が選択された。または画面切り替えの必要がある。

wParam

■ Hwnd ウィンドウハンドル

lParam (HIWORD)

■ ID MENUWHOLE 構造体で指定した ID 番号

lParam (LOWORD)

■ CTL_UP_OVER 前ページに移動しました。

■ CTL_UP_SCROLLOVER 前項目(上矢印)に移動しました。

■ CTL_DOWN_SCROLLOVER 次項目(下矢印)に移動しました。

■ CTL_DOWN_OVER 次ページに移動しました。

■ CTL_KEY_00 0 番キーが押下されました。

■ CTL_KEY_01 1 番キーが押下されました。

■ CTL_KEY_02 2 番キーが押下されました。

■ CTL_KEY_03 3 番キーが押下されました。

■ CTL_KEY_04 4 番キーが押下されました。

■ CTL_KEY_05 5 番キーが押下されました。

■ CTL_KEY_06 6 番キーが押下されました。

■ CTL_KEY_07 7 番キーが押下されました。

■ CTL_KEY_08 8 番キーが押下されました。

■ CTL_KEY_09 9 番キーが押下されました。

■ CTL_KEY_CLEAR クリアキーが押下されました。

■ CTL_KEY_ENTER エンターキーが押下されました。

(カーソル移動できない場合、エンター押下時設定される。)

UM_CTLCHECK

メニューコントロール内のチェックボックスの状態が変更された。

wParam

■ Hwnd ウィンドウハンドル

lParam (HIWORD)

■ ID MENUWHOLE 構造体で指定した ID 番号

lParam (LOWORD)

■ 1~9 操作されたチェックボックスの行番号

CTLCreateBaseWindow

機能 DT950 のメニューを表示しない画面を作成する場合に使用します。
アプリから親ウィンドウのハンドル、タイトル文字列、ウィンドウクラス名などを指定して Create します。不要になったら Close します。
パラメータはメニューと同じで、タイトルや各行の文字列を指定すれば表示を作れます。
Windows 標準のダイアログボックスのようなものです。ここにビットマップを貼り付けたり、エディットコントロールなどをおくことができます。
このウィンドウに対するメッセージは、アプリが指定したウィンドウクラスのウィンドウプロシージャに送られます。

書式 **HWND CTLCreateBaseWindow(HWND hwnd,**
LPCTSTR lpszClassName,
HINSTANCE hInst,
MENUWHOLE *pmw,
MENUITEM *pmi
);

パラメータ

hWnd : Create するウィンドウの親ウィンドウハンドルを指定します。

LpszClassName : 登録されたクラス名のアドレスを指定します。
(このウィンドウクラスのプロシージャにメッセージ送られます)

hInst : アプリケーションインスタンスのハンドルを指定します。

pmw : MENUWHOLE 構造体のポインタを指定します。

pmi : MENUITEM 構造体のポインタを指定します。

戻り値

関数が成功すると、作成されたウィンドウのハンドルが返ります。

関数が失敗すると、NULL が返ります。

CTLCloseBaseWindow

機能 ベースウィンドウを破棄します。

書式 `bool CTLCloseBaseWindow(HWND hwnd);`

パラメータ

hwnd : 破棄するウィンドウハンドルを指定します。

戻り値

TRUE	正常終了
FALSE	エラー

CTLMessageBox

機能 表示されている画面上に、メッセージを表示したい場合に使用します。
パラメータにより、タイトル、メッセージ、メッセージに対するキーによる確認動作を数タイプから選択することが可能です。

書式 `int CTLMessageBox(HWND hWnd,
LPCTSTR lpText,
LPCTSTR lpCaption,
UINT uType
);`

パラメータ

hWnd : 作成したいメッセージボックスのオーナーウィンドウのハンドルを指定します。NULLを指定すると、オーナーウィンドウを持たないメッセージボックスが作成されます。

lpText : 表示したいメッセージを NULL で終わる文字列へのポインタを指定します。

複数行指定する場合は、文字列に改行コード(\n)を含めて下さい。

(メッセージのみの場合で、最高9行 (全角90文字まで))

lpCaption : メッセージボックスのタイトルを、NULLで終わる文字列へのポインタを指定します。

単行のみ (全角10文字、半角20文字まで)

uType : メッセージに対するキーによる確認動作を、次頁のいずれかのパターンより設定します。

uType 値

値	意味
CTL_M_YESNO	1. はい 2. いいえを表示し, 1. はい がデフォルトです
CTL_M_YESNO2	1. はい 2. いいえを表示し, 2. いいえ がデフォルトです
CTL_M_YESNOCANCEL	1. はい 2. いいえ 3. キャンセルを表示し, 1. はい がデフォルトです
CTL_M_YESNOCANCEL2	1. はい 2. いいえ 3. キャンセルを表示し, 2. いいえ がデフォルトです
CTL_M_YESNOCANCEL3	1. はい 2. いいえ 3. キャンセルを表示し, 3. キャンセル がデフォルトです
CTL_M_OK	1. OK を表示します。
CTL_M_CANCEL	1. キャンセルを表示します
CTL_M_CANCELTRYCONTINUE	1. キャンセル 2. 再実行 3. 続行を表示し, 1. キャンセル がデフォルトです
CTL_M_CANCELTRYCONTINUE2	1. キャンセル 2. 再実行 3. 続行を表示し, 2. 再実行 がデフォルトです
CTL_M_CANCELTRYCONTINUE3	1. キャンセル 2. 再実行 3. 続行を表示し, 3. 続行 がデフォルトです
CTL_M_RETRYCANCEL	1. 再試行 2. キャンセルを表示し, 1. 再試行 がデフォルトです
CTL_M_RETRYCANCEL2	1. 再試行 2. キャンセルを表示し, 2. キャンセル がデフォルトです
CTL_M_NONE	メッセージ+最下部に [ENT] 閉じる と表示します
CTL_M_NONE_CLOSE	メッセージのみ表示します
CTL_M_HELP	メッセージ+最下部に [ENT] 閉じる と表示します
CTL_M_HELP_CLOSE	メッセージのみ表示します

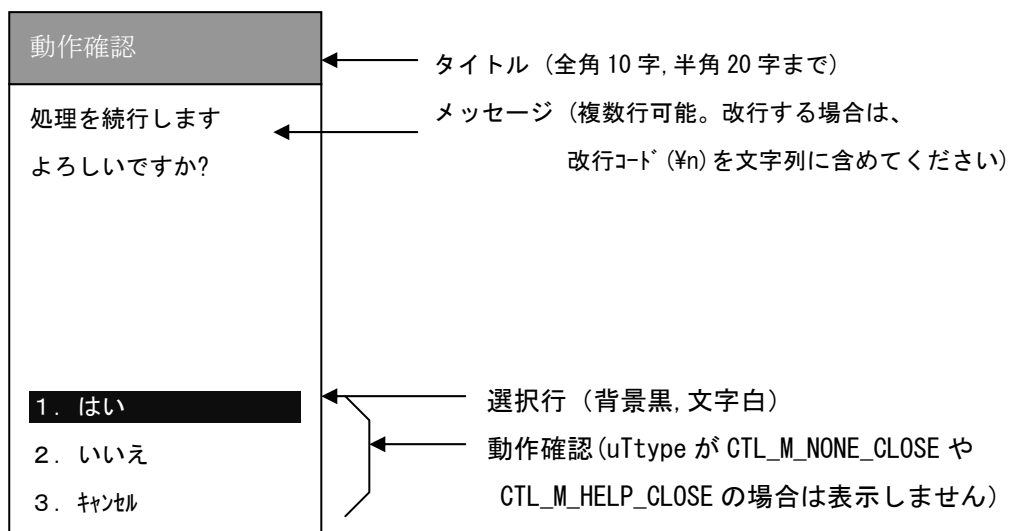
※CTL_M_NONE と CTL_M_HELP の違い(CLOSE も含む)は CTL_M_HELP が[L]キーおよび[R]キーも有効であることです。

※CTL_M_NONE と CTL_M_NONE_CLOSE の違い(HELP も含む)は CTL_M_NONE_CLOSE が最下部行に” [ENT] 閉じる”を表示しません。

上記パラメータに次のフラグを組み合わせて指定することもできます。

値	意味
CTL_M_TOPMOST	最前面ウィンドウとして作成します。
CTL_M_YESNO2	作成時、メッセージボックスをフォアグラウンドにします。

表示例 (uType に M_YESNOCANCEL を指定した場合)



※ 表示メッセージの文字列が長く、表示しきれない場合は途中でカットされます。
(表示上限文字数は引数 uType により異なります。)

戻り値

関数が成功すると、次頁の値のいずれかが返ります。失敗すると、0 (FALSE)が返ります。

値	意味
CTL_I_CANCEL	[キャンセル] に相当するキーが押されました。
CTL_I_CONTINUE	[続行] に相当するキーが押されました。
CTL_I_NO	[いいえ] に相当するキーが押されました。
CTL_I_OK	[OK] に相当するキーが押されました。
CTL_I_RETRY	[再試行] に相当するキーが押されました。
CTL_I_TRYAGAIN	[再実行] に相当するキーが押されました。
CTL_I_YES	[はい] に相当するキーが押されました。
CTL_I_LEFT	L キーが押されました。
CTL_I_RIGHT	R キーが押されました。

- ※ 数字キー, 上下キー, Enter キー, マルチキー, CLR キー (含 Fn 同時), L キー, R キー以外が押下された場合は無視します。(uType の設定により異なります)
- ※ 上下キーはカーソルが移動するのみで本ルーチンからは抜けません。
- ※ CLR および Fn+CLR キーが押下されると CTL_I_CANCEL を返します。
- ※ uType が CTL_M_NONE および CTL_M_NONE_CLOSE の場合、Enter キーが押下された場合は CTL_I_OK を返します。
- ※ uType が CTL_M_HELP および CTL_M_HELP_CLOSE の場合、Enter キーが押下された場合は CTL_I_OK を返し、L キーならば CTL_I_LEFT, R キーならば CTL_I_RIGHT を返します。

注意

- ※ このコントロール呼び出し以前にアプリでフォーカスを持っていた場合、呼び出し後に必ず SetFocus() 等でフォーカスを呼び出し前に戻すようにして下さい。(参考: 2-2)

CTLCreateEditControl

機能 文字入力をさせたい場合に使用します。
Windows 標準のエディットコントロールとの違いは、CLR キーや上下カーソルキーなど特定のキー入力があった時にアプリに通知させることができます。

書式 `HWND CTLCreateEditControl(DWORD dwExStyle,
LPCTSTR lpszClassName,
LPCTSTR lpszWindowName,
DWORD dwStyle,
int x,
int y,
int nWidth,
int nHeight,
HWND hwndParent,
HMENU hmenu,
HINSTANCE hinst,
LPVOID lpvParam,
HWND hwndMessage,
DWORD dwFlag
);`

パラメータ

dwExStyle : 拡張ウインドウスタイル

lpszClassName : 登録されたクラス名のアドレス (TEXT("EDIT"))

lpszWindowName : ウインドウ名のアドレス

dwStyle : ウインドウスタイル

x : ウインドウの水平座標の位置

y : ウインドウの垂直座標の位置

(タイトルバーは高さ 16 ドット[座標 0-15]なので 16 以上を指定してください)

nWidth : ウインドウの幅

(1 文字のサイズは全角 12 × 12 ドット、半角 6 × 12 ドットです)

nHeight : ウインドウの高さ

hwndParent : 親ウインドウまたはオーナーウインドウのハンドル

hmenu : メニューのハンドルまたは子ウインドウの ID

hinst : アプリケーションインスタンスのハンドル

lpvParam : ウインドウ作成データのアドレス

hwndMessage : dwFlag で指定したキーが押された時にメッセージを送信するウインドウ

dwFlag : アプリケーションに通知するキー

このパラメータは、次のページの組み合わせになります。

値	意味
CTL_KEYUP	上カーソルキーが押されました。
CTL_KEYDOWN	下カーソルキーが押されました。
CTL_KEYENTER	エンターキーが押されました。
CTL_KEYCLR	クリアキーが押されました。
CTL_KEYESC	ESC キーが押されました。
CTL_KEYFNNUM	Fn+数字キーが押されました。
CTL_KEYFNUP	Fn+上カーソルキーが押されました。
CTL_KEYFNDOWN	Fn+下カーソルキーが押されました。
CTL_KEYFNLEFT	Fn+左カーソルキーが押されました。
CTL_KEYFNRIGHT	Fn+右カーソルキーが押されました。

dwFlag で指定したキーが押されると、hwndMessage で指定したウインドウにメッセージ UM_CTLEDIT が送信されます。

戻り値

関数が正常に終了した場合は、作成したエディットコントロールのハンドルを返します。それ以外の場合は、NULL を返します。

備考

戻り値のハンドルに対して WM_SETTEXT、WM_GETTEXT、EM_LIMITTEXT などのメッセージを送信することにより、エディットコントロールの制御が行えます。

dwFlag で指定しないキーの動作はオリジナルのエディットボックスの動作に従います。

CLR キーはボックス内に入力文字列がある場合は全文字を消去します。

ボックスが空の場合は、dwFlag で CTL_KEYCLR が指定されている場合は親ウインドウにメッセージを送信し、dwFlag で CTL_KEYCLR が指定されていない場合は NOP になります。

エディットコントロール内のコピー機能はありません。

ES_MULTILINE スタイルを指定したエディットボックスでも CTL_KEYUP や CTL_KEYDOWN や CTL_KEYENTER を指定することができますが、エディットボックス内でのカーソルの上下移動や改行入力ができなくなりますので注意してください。

CTLCloseEditControl

機能 エディットコントロールを破棄します。

書式 `BOOL CTLCloseEditControl(HWND hwndEdit);`

パラメータ

hwndEdit : 破棄するウィンドウハンドルを指定します。

戻り値

TRUE	正常終了
FALSE	エラー

エディットボックスで利用されるメッセージ

UM_CTLEDIT

エディットボックスで指定されたキー入力があった。

wParam

■ HWND キー入力があったエディットボックスのハンドル

lParam

■ CTL_KEYUP	上カーソルキーが押されました。
■ CTL_KEYDOWN	下カーソルキーが押されました。
■ CTL_KEYENTER	エンターキーが押されました。
■ CTL_KEYCLR	クリアキーが押されました。
■ CTL_KEYESC	ESC キーが押されました。
■ CTL_KEYFN0	Fn+0 が押されました。
■ CTL_KEYFN1	Fn+1 が押されました。
■ CTL_KEYFN2	Fn+2 が押されました。
■ CTL_KEYFN3	Fn+3 が押されました。
■ CTL_KEYFN4	Fn+4 が押されました。
■ CTL_KEYFN5	Fn+5 が押されました。
■ CTL_KEYFN6	Fn+6 が押されました。
■ CTL_KEYFN7	Fn+7 が押されました。
■ CTL_KEYFN8	Fn+8 が押されました。
■ CTL_KEYFN9	Fn+9 が押されました。
■ CTL_KEYFNUP	Fn+上カーソルが押されました。
■ CTL_KEYFNDOWN	Fn+下カーソルが押されました。
■ CTL_KEYFNLEFT	Fn+左カーソルが押されました。
■ CTL_KEYFNRIGHT	Fn+右カーソルが押されました。

※一部の Fn+数字キーにはバックライト ON/OFF などの機能が割り当てられています。

パラメータは CTL_KEYFN0 から CTL_KEYFN9 まで用意してありますが、機能付きのキーが押されてもアプリには通知されません。

CTLFileDialogControl

機能 DT950 のファイル選択を行う場合に使用します。
読込用ファイルの指定、保存用ファイルの指定、フォルダの選択が可能です。
また、指定されたフォルダより上位への移動を禁止することも可能です。

書式 **BOOL CTLFileDialogControl(HWND hWnd,**
HINSTANCE hInstance,
LPCTSTR pDefaultFolder,
INT dwOperateMode,
LPCTSTR pReturnFilePath,
LPCTSTR pMaskExtension
);

パラメータ

hWnd : Create するウィンドウの親ウィンドウハンドルを指定します。

hInstance : 親ウィンドウのインスタンスを指定します。

pDdefaultFolder : ウィンドウオープン時の初期フォルダ位置を指定します。(必須)

dwOperateMode : 実行時の動作を指定します。

以下の値が指定できます。

値	意味
CTL_FD_FILEOPEN	ファイルを開くモードで動作する。
CTL_FD_FILESAVE	ファイルへ保存するモードで動作する。
CTL_FD_FOLDERSELECT	フォルダ選択を行う。(ファイルは表示されなくなります)
CTL_FD_CHFOLDERROOT	pDefaultFolder で指定されたフォルダより上位への移動を禁止します。
CTL_FD_IEFAVORITESADD	IE 用のお気に入り追加を行う場合の専用モードです。
CTL_FD_IEFAVORITESSEL	IE 用のお気に入りの選択を行う場合の専用モードです。
CTL_FD_IEFAVORITESTO	IE 用のお気に入りの移動先選択を行なう場合の専用モードです。

※ CTL_FD_IEFAVORITESADD、CTL_FD_IEFAVORITESSEL、CTL_FD_IEFAVORITESTO の3つは、IE 専用です。

pReturnFilePath : ユーザが選択したファイル名を保存するエリアのポインタを指定します。

(必須) このエリアにデフォルトとなるファイル名を入れておくことが可能です。

※この内容は、正しく設定して下さい。不定な値のままの場合は、メニュー

よりファイル名入力を選択した際に、不正な表示が行われる場合があります。

pMaskExtension : ファイルの拡張子で表示されるファイルを制限する場合に、その拡張子を

指定します。指定しない場合は、NULL を指定します。

※ 現在指定できる拡張子は、ひとつのみです。半角 3 文字で指定してください。

大文字小文字は、同一として判定されます。

戻り値

ユーザがファイルまたはフォルダを指定すると、TRUE が返ります。

ユーザが選択を行わずに終了すると、FALSE が返ります。

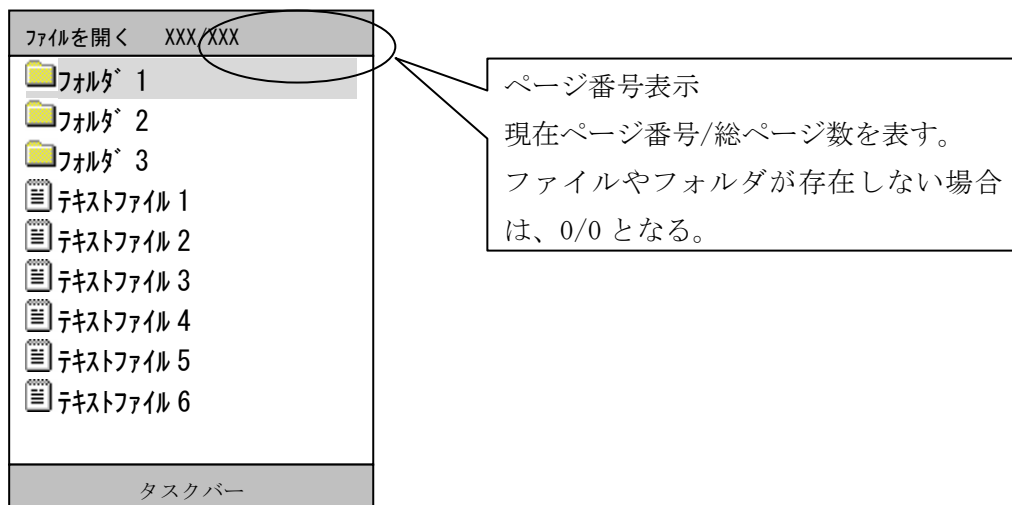
- ※ ユーザが選択したファイル名が pReturnFilePath で指定されたエリアに格納されます。
ファイルおよびフォルダの存在性チェックは、ダイアログ側では行っていません。
必ずアプリケーションにて、チェックし必要な処理を行ってください。

注意

- ※ このコントロール呼び出し以前にアプリでフォーカスを持っていた場合、
呼び出し後に必ず SetFocus() 等でフォーカスを呼び出し前に戻すようにして下さい。
(参考：2-2)

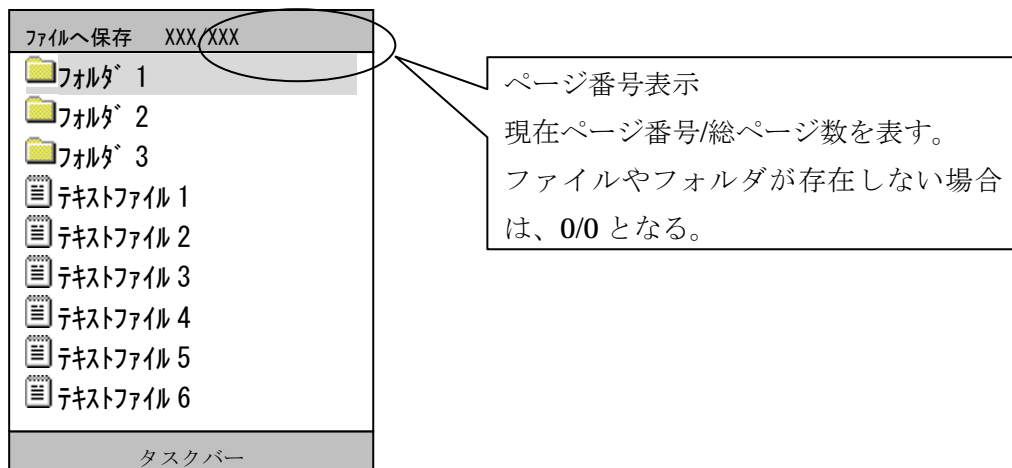
ファイルダイアログの表示画面

・ CTL_FD_FILEOPEN で呼び出した時の画面



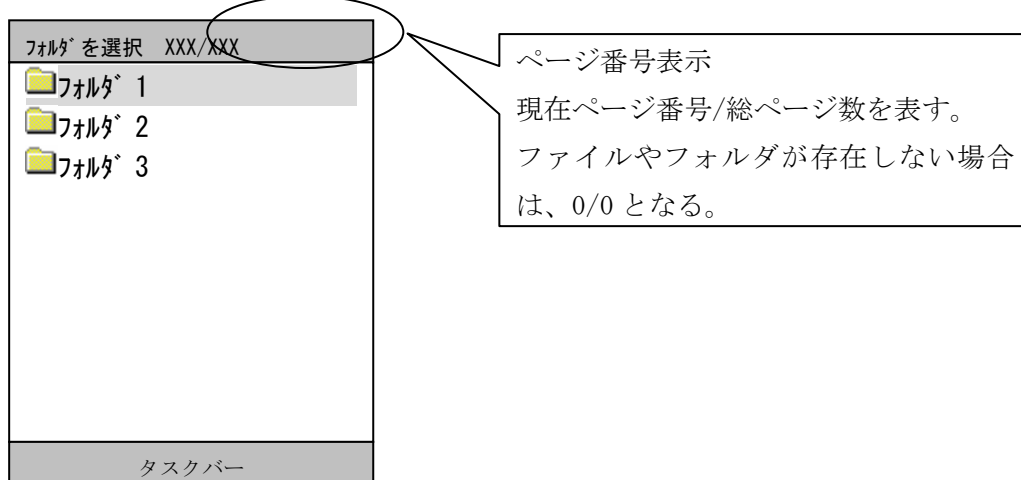
- [R]キー押下で、次のページへ（存在しない場合は変化せず）
- [L]キー押下で、前のページへ（存在しない場合は変化せず）
ファイルを選択時、Enter キー押下にて、そのファイルを選択し、FileDialogControl を終了する。
- [↑]キー押下で、選択をひとつ上の行へ（存在しない場合は変化せず、ただし前ページがある場合は、前ページの最終行へ）
- [↓]キー押下で、選択をひとつ下の行へ（存在しない場合は変化せず、ただし次ページがある場合は、次ページの先頭行へ）
フォルダ選択時、[Fn]+[R]キー押下でそのフォルダ内へ
- [Fn]+[L]キー押下時、上位フォルダが存在する場合は、上位フォルダへ（ただし、現在のフォルダより上位フォルダへの移動を禁止されている場合は、変化せず）
- [F8]（メニュー）キー押下時、メニュー画面を表示する。

・ CTL_FD_FILESAVE で呼び出した時の画面



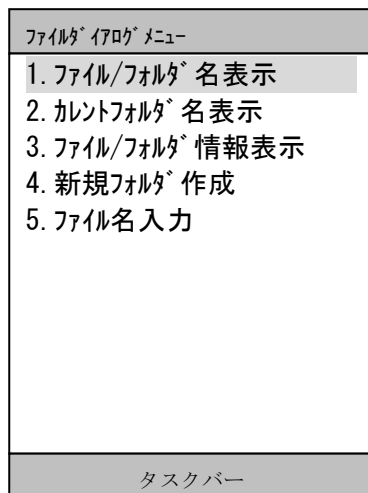
- [R]キー押下で、次のページへ（存在しない場合は変化せず）
- [L]キー押下で、前のページへ（存在しない場合は変化せず）
ファイルを選択時、Enter キー押下にて、そのファイルを選択し、FileDialogControl を終了する。
- [↑]キー押下で、選択をひとつ上の行へ（存在しない場合は変化せず、ただし前ページがある場合は、前ページの最終行へ）
- [↓]キー押下で、選択をひとつ下の行へ（存在しない場合は変化せず、ただし次ページがある場合は、次ページの先頭行へ）
フォルダ選択時、[Fn]+[R]キー押下でそのフォルダ内へ
- [Fn]+[L]キー押下時、上位フォルダが存在する場合は、上位フォルダへ（ただし、現在のフォルダより上位フォルダへの移動を禁止されている場合は、変化せず）
- [F8] (メニュー)キー押下時、メニュー画面を表示する。

・ CTL_FD_FOLDERSELECT で呼び出した時の画面



- [R]キー押下で、次のページへ（存在しない場合は、先頭ページへ）
- [L]キー押下で、前のページへ（存在しない場合は、最終ページへ）
フォルダを選択時、Enter キー押下にて、そのフォルダを選択し、FileDialogControl を終了する。
- [↑]キー押下で、選択をひとつ上の行へ（存在しない場合は、前ページ又は、最終ページの最後の行へ）
- [↓]キー押下で、選択をひとつ下の行へ（存在しない場合は、次ページ又は、先頭ページの最初の行へ）
フォルダ選択時、[Fn]+[R]キー押下でそのフォルダ内へ
- フォルダ選択時、[Fn]+[L]キー押下時、上位フォルダが存在する場合は、上位フォルダへ（ただし、現在のフォルダより上位フォルダへの移動を禁止されている場合は、変化せず）
- [F8] (メニュー)キー押下時、メニュー画面を表示する。

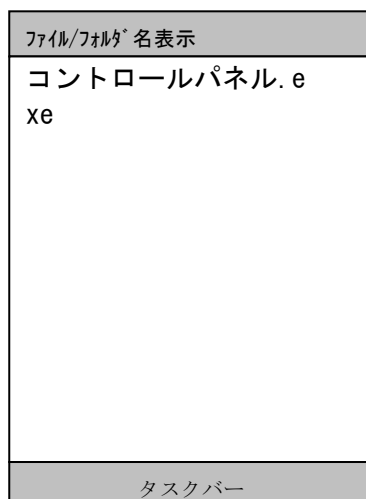
・メニューボタンを押下した時の画面



各メニューの動作について

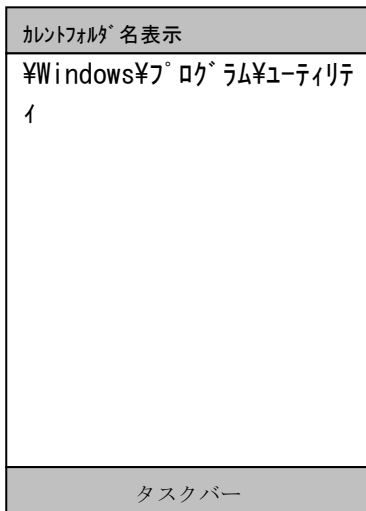
1. ファイル/フォルダ名表示
現在選択されているファイルまたはフォルダの名称を1画面に表示する。
2. カレントフォルダ名表示
現在対象となっているフォルダ位置を1画面に表示する。
3. ファイル/フォルダ情報表示
現在選択されているファイルまたはフォルダの情報を1画面に表示する。
4. 新規フォルダ作成
カレントフォルダ上に、指定された名前のフォルダを作成する。
5. ファイル名入力
対象とするファイルの名称を直接入力する。
フォルダ選択時は、本メニューは選択不可とする。
CLR キーでファイル一覧に戻る。

・メニューにてファイル/フォルダ名表示を選択した時の画面



Enter キーで一覧画面へ戻る。

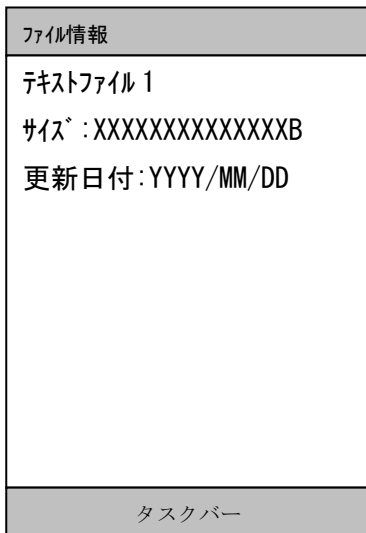
・メニューにてカレントフォルダ名表示を選択した時の画面



Enter キーで一覧画面へ戻る。

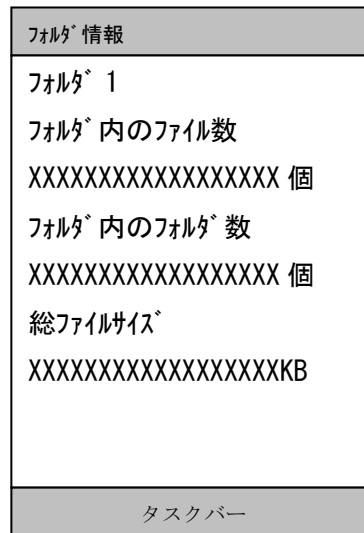
・メニューにてファイル/フォルダ情報表示を選択した時の画面

ファイルの時

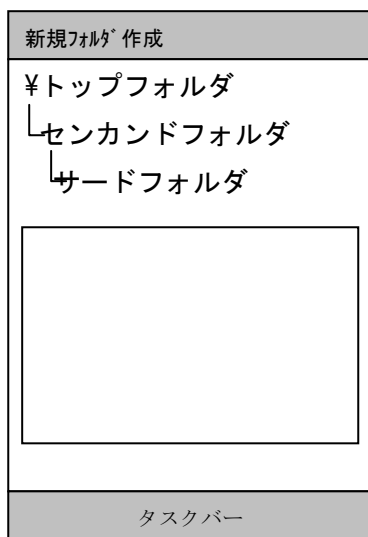


Enter キーで一覧画面へ戻る。

フォルダの時



・メニューにて新規フォルダ作成を選択した時の画面

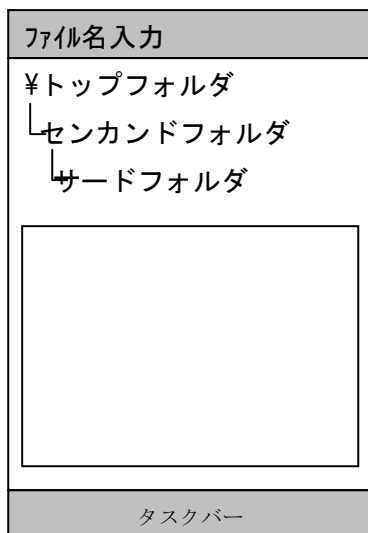


フォルダ名を入力して Enter で新規フォルダを作成し、一覧画面へ戻る。

IE お気に入り追加および IE お気に入り選択時は、フォルダが選択されている場合は、そのフォルダ以下に新規フォルダを作成します。それ以外は、カレントのフォルダに作成します。

CLR キー押下時、フォルダ作成を中止して、一覧画面へ戻る。

・メニューにてファイル名入力を選択した時の画面



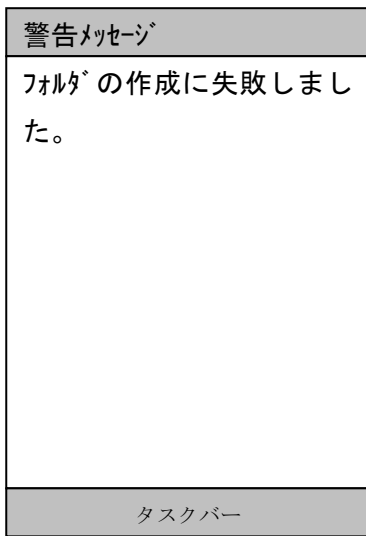
ファイル名を入力して Enter で FileDialogControl を終了する。

CLR キーにて、ファイル一覧へ戻る。

FILEOPEN および FILESAVE で、デフォルトファイル名が指定されている場合は、デフォルトファイル名を表示します。デフォルトファイル名がない場合で、ファイルが選択されている時は、選択されたファイル名を表示します。IE お気に入り追加および IE お気に入り選択の場合は、フォルダが選択されている場合は、そのフォルダ以下として、ファイル名を作成します。それ以外は、カレントフォルダにファイル名を作成します。

※本画面は、FILESAVE 時および IE お気に入り追加機能において最終的なファイル名を確認するために必ず表示されます。確認にて ENTER 後、FileDialogControl を終了します。

- ・新規フォルダ作成時にフォルダが作成できなかった時の画面



Enter キーでファイル一覧に戻る。

CTLCreateMenuControlRC

機能 リソース ID を指定してメニューコントロールの作成ができます。
機能は CTLCreateMenuControl と同じで、MENUITEM 構造体の代わりにリソース ID を指定します。
MENUWHOLE 構造体の IID パラメータは4つ目のパラメータ IID が設定されるので、設定不要です。

書式 `HWND CTLCreateMenuControlRC(HWND hwnd,
MENUWHOLE *pmw,
HINSTANCE hInstance,
int IID
);`

パラメータ

hwnd : 親ウィンドウのハンドル

pmw : MENUWHOLE 構造体

hInstance : アプリケーション インスタンスのハンドル

IID : リソース ID

戻り値

関数が成功すると、作成されたウィンドウのハンドルが返ります。
関数が失敗すると、NULL が返ります。

注意

この関数は、リソースファイルを利用するので、VB、C#からは利用できません。

CTLSetMenuParamRC

機能 リソース ID を指定してメニュー画面の変更ができます。
機能は CTLSetMenuParam と同じで、MENUITEM 構造体の代わりにリソース ID を指定します。

書式 `bool CTLSetMenuParamRC(HWND hwnd,
MENUWHOLE *pmw,
HINSTANCE hInstance,
int IID
);`

パラメータ

hwnd : 値を設定するメニューウインドウのハンドル

pmw : MENUWHOLE 構造体

hInstance : アプリケーション インスタンスのハンドル

IID : リソース ID

戻り値

TRUE 正常終了

FALSE エラー

注意

この関数は、リソースファイルを利用するので、VB、C#からは利用できません。

CTLGetMenuRC

機能 リソースで作成した情報を MENUWHOLE 構造体と MENUITEM 構造体に読み込みます。表示文字列が可変の場合などは、固定部分をリソースで持ち、この関数でリソースデータを読み込み、文字列を編集して CTLCreateMenuControl 関数でメニューを作ります。

書式

```
bool CTLGetMenuRC( MENUWHOLE *pmw,  
                  MENUITEM *pmi,  
                  HINSTANCE hInstance,  
                  int iID,  
                  LPCTSTR lpszTitle,  
                  DWORD dwSizeT,  
                  LPCTSTR lpszLabel,  
                  DWORD dwSizeL  
                  );
```

パラメータ

pmw : MENUWHOLE 構造体のポインタを指定します。

pmi : MENUITEM 構造体のポインタを指定します。

hInst : アプリケーションインスタンスのハンドルを指定します。

iID : リソースの ID を指定します。

lpszTitle : アプリ側が用意するタイトル文字列を格納する変数のポインタを指定します。

dwSizeT : 上記変数のサイズ (ワード単位) を指定します。

lpszLabel : アプリ側が用意する各行のラベル文字列を格納する変数のポインタを指定します。

dwSizeL : 上記変数のサイズ (ワード単位) を指定します。

戻り値

TRUE	正常終了
FALSE	エラー

注意

この関数は、リソースファイルを利用するので、VB、C#からは利用できません。

CTLCreateBaseWindowRC

機能 リソース ID を指定してベースウインドウの作成ができます。
機能は CTLCreateBaseWindow と同じで、MENUITEM 構造体の代わりにリソース ID を指定します。

書式 `HWND CTLCreateBaseWindowRC(HWND hwnd,
LPCTSTR lpszClassName,
HINSTANCE hInstance,
int iID
);`

パラメータ

hwnd : 親ウインドウのハンドル

lpszClassName : ウィンドウクラス名

hInstance : アプリケーション インスタンスのハンドル

iID : リソース ID

戻り値

関数が成功すると、作成されたウインドウのハンドルが返ります。

関数が失敗すると、NULL が返ります。

注意

この関数は、リソースファイルを利用するので、VB、C#からは利用できません。

CTLDispWaitCursor

機能 擬似的な砂時計表示を行います。

書式 `void CTLDispWaitCursor(void);`

パラメータ

ありません。

戻り値

ありません。

CTLUndispWaitCursor

機能 擬似的な砂時計表示を破棄します。

書式 `void CTLUndispWaitCursor(void);`

パラメータ

ありません。

戻り値

ありません。

リソースを使用する場合

リソースエディタで Menu を作成し、リソース ID を引数としてメニューコントロールを作成することができます。

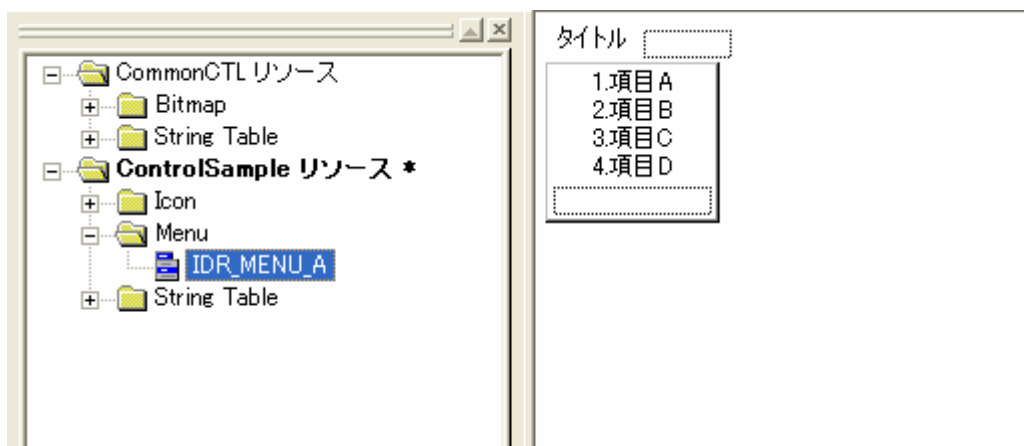
< !ビルド時の注意! >

CTL_KEY_01~CTL_KEY_09 や CTL_NO_KEY_VALUE は CTLDefine.h で define 定義されています。

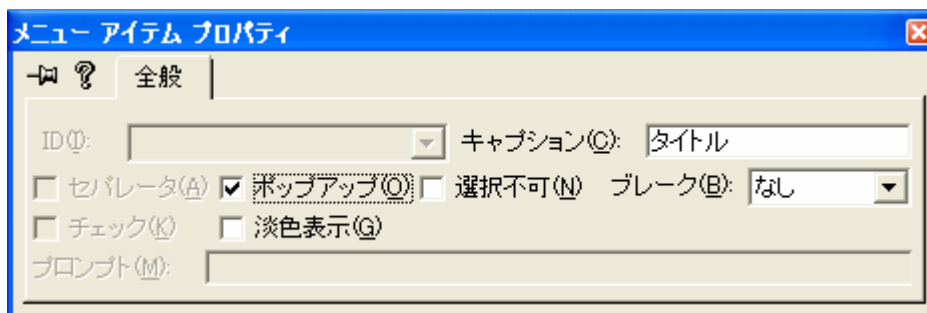
リソースエディタに番号をふられると値が一致なくなってしまう。

resource.h に CTL_KEY_01 などが生成されてしまった場合は、それを削除して、newres.h に「#include "CTLDefine.h"」を記述するなどしてラベルの重複をさけてください。

embedded Visual C++ のリソースエディタで Menu を作成します。

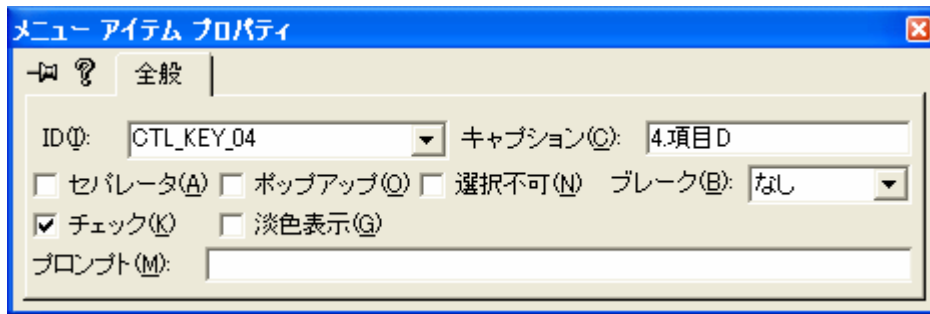


・タイトルの入力



最初のメニューアイテムプロパティの「キャプション」に入力したものが、メニュー画面のタイトルになります。これは MENUWHOLE 構造体の lpTitle にあたります。

・メニュー項目の入力



MENUITEM 構造体の各項目を入力します。

lpLabel は「キャプション」に入力します。

iKeyVal は「ID」に入力します。

bCursor も「ID」で指定します。

ID に CTL_KEY_01～CTL_KEY_09 を指定するとカーソル選択有効、CTL_NO_KEY_VALUE を指定するとカーソル選択無効になります

bActive は「淡色表示」で指定します。

チェックしない場合は通常の文字表示、チェックするとグレー表示になります

bCheck は「チェック」で指定します。

チェックしない場合は通常の項目、チェックするとチェックボックスがつきます

・リソースファイル

必要な項目を入力して SAVE すると、例えば下記のようなリソースが出力されます。

```
IDR_MENU1 MENU DISCARDABLE
BEGIN
  POPUP "タイトル"
  BEGIN
    MENUITEM "1. 項目 A", CTL_KEY_01
    MENUITEM "2. 項目 B", CTL_KEY_02, CHECKED
    MENUITEM "3. 項目 C", CTL_KEY_03, GRAYED
    MENUITEM "4. 項目 D", CTL_KEY_04
  END
END
END
```


カシオ計算機お問い合わせ窓口

平成 17 年 12 月現在

製品のご購入についてのご相談

システムソリューション営業統轄部 TEL03-5334-4638(代)

〒151-8543 東京都渋谷区本町 1-6-2

法人第一営業部 TEL03-5334-4534

〒151-8543 東京都渋谷区本町 1-6-2

法人第二営業部 TEL03-5334-4447

〒151-8543 東京都渋谷区本町 1-6-2

西日本営業部 TEL06-6243-1717

〒541-0056 大阪市中央区久太郎町 3-6-8 御堂筋ダイワボウビル 7F

名古屋営業所 TEL052-324-2104

〒460-0024 名古屋市中区正木 3-9-27 NFC 金山ビル 4F

ソフトウェア開発に関するお問い合わせ

カシオデベロッパースポートサービス(有償)

<https://m-techinfo.casio.co.jp/free/index.php>

製品に関する最新情報

カシオテクニカルインフォメーションサービス

<https://m-techinfo.casio.co.jp/free/index.php>

製品の取扱い方法のお問い合わせ

情報機器コールセンター



0570-022066

市内通話料金でご利用いただけます。

携帯電話・PHS 等をご利用の場合、03-5294-7251

カシオ計算機株式会社

〒151-8543 東京都渋谷区本町 1-6-2

TEL 03-5334-4638(代)