

ル・クロン
K-tai Developer
for BREW

リリースノート (BREW3.1対応版)

第1.0版

株式会社ソア・システムズ

はじめに

本書は、ル・クロン K-tai for BREW製品のBREW3.1対応版に対し、新しく追加・変更された機能、および未対応となった機能について説明しております。

なお、本書記載以外の機能については、専用API説明書_BREW、および文法解説書・ヘルプをご参照ください。

ご注意

本書の内容の一部、または全部を無断で転載することは、固くお断りいたします。
本書の内容に付きましては、将来予告なしに変更する場合があります。
本書の内容に付きましては万全を期して作成いたしましたが、万一誤り・お気付きの点がありましたら、ご連絡下さいますようお願いいたします。

2005.10.14更新

CONTENTS

目次

1. 概要	5
1.1. 追加された機能	5
1.2. 変更された機能	5
1.3. 未対応となった機能	5
2. 新機能の説明	6
(1) 携帯電話に合わせたフォーカス移動と入力方式	6
(2) 動的コントロール	7
(3) コントロール名の引数による代入、参照関数の追加	8
(4) リストコントロール	8
(5) プロジェクトプロパティ	10
(6) 画像のリソース化	11
(7) 画像表示の拡張 (透過、拡大縮小、絵文字化、動的な表示)	12
(8) 現在および一つ前のフォーカスのコントロール名取得	12
(9) ButtonAction の「CALL」の拡張	12
(10) コントロールの形状の拡張 (角丸、端丸等)	13
(11) ドロップダウンリストの拡張 (表示行数、フォントの指定が可能)	14
(12) スクロールバーの拡張 (幅、ドラッグ動作、リピートの指定が可能)	14
(13) フォントの拡張 (ボールド)	14
(14) 入力方式の拡張 (上書きモード、電卓入力、文字チェック、数値の桁数指定)	14
(15) テキストコントロールの拡張 (MASK、Alignment)	14
(16) フォーム背景への画像貼付け	14
(17) イベント追加 (イメージコントロールに \$Click イベント)	14
(18) カレントフォーカスの色指定	14
3. 関数リファレンス	16
3.1. フォーム関連関数 (共通)	17
■ CreateArray (クワイト・アレイ) コントロールの生成	17
■ CurFocus (カレント・フォーカス) 現在のフォーカス情報取得	19
■ PrevFocus (プレVIOUS・フォーカス) 一つ前のフォーカス情報取得	20
■ GetValueByName (ゲットバリュー・バイネーム) 名前によるコントロール値取得	21
■ SetValueByName (セットバリュー・バイネーム) 名前によるコントロール値設定	22
■ RscAlloc (リソース・アロケート) リソース領域の確保	23
■ RscSetProp (リソース・セット・プロパティ) リソースプロパティの設定	24
■ RscFree (リソース・フリ) リソース領域の開放	24
■ SetCrtCrsClr (セット・カレントカラー・カラー) カレントフォーカス色の設定	25
■ ScrollList (スクロール・リスト) リスト行スクロール	26
■ ListDispItem (リスト・ディスプレイ・アイテム) リスト表示アイテム切り替え	27
3.2. 携帯電話専用関数 (携帯共通)	28
■ SetControlTab フォーカス移動順序設定	28
■ CursorKeyMode (カーソルキーモード) 移動キー設定	29
■ ChgCancelKey (チェンジ・キャンセルキー) キャンセルキー変更	30
3.3. 携帯電話専用関数 (機種専用)	31

1. 概要

1.1. 追加された機能

- (1) DB の破損チェックと破損した場合のリカバリ機能 ([ISMRECOV](#))
- (2) データフォルダ内のファイル操作機能
[IdataFolder](#)、[IdataFolder_ENUMInit](#)、[IdataFolder_ENUMNext](#)、[IdataFolder_FileInfo](#)、[IdataFolder_Get](#)、[IdataFolder_Pet](#)、[IdataFolder_Free](#)、[IdataFolder_SelectUI](#)、[IdataFolder_SelectNM](#)、[IdataFolder_DeleteUI](#)
- (3) ヒープメモリ取得機能 ([GetRAMFree](#))
- (4) バックライトの制御機能 ([BackLight](#))
- (5) 電界強度取得機能 ([Intensity](#))

1.2. 変更された機能

- (1) プロパティ変更
処理式でプロパティ変更する際の桁数制約の関係で下記プロパティ名が変更されました。
[InActivCBackColor](#) [InActCBackColor](#) に変更。
[InActCForeColor](#) [InActCForeColor](#) に変更。
- (2)

1.3. 未対応となった機能

- (1) アドレス帳呼び出し ([AddrBook](#))
BREW3.1 のアドレス帳は BREW アプリに変更となりインタフェースが全く異なるため当面未対応となります。
- (2)

2.新機能の説明

(1) 携帯電話に合わせたフォーカス移動と入力方式

ペンタッチができないという携帯電話の操作性を考慮し、以下の機能が追加されています。

フォーカス移動については、ラベル、イメージを含む全てのコントロールが移動対象となります。コントロールのプロパティで「フォーカスを受け取る」と設定しておくことで、自動で上下移動キー()によるフォーカス移動がサポートされます。
 なお、移動対象としたくない場合は、「フォーカスを受け取らない」にしておきます。

フォーカスの移動方向や移動キーは **SetControlTab**、**CursorKeyMode** 関数で設定変更することができます。

- ・移動モード：SetControlTab 関数
 - 1：上 下 左 右順 (規定値)
 - 2：左 右 上 下順
- ・移動キー：CursorKeyMode 関数
 - 0： キーで移動 (規定値)
 - 1： キーで移動
 - 2： キーで上下左右に移動 左右に近いコントロール。
上下の近いコントロールに移動 (左上を起点)

テキストコントロールに「フォーカスがあるが入力可能でない状態」というモードが追加されました。携帯電話では、入力可能なテキストコントロールにフォーカスが移動しただけでは通常の入力可能状態にはなりません。

フォーカスのある状態で「Center (Select) キー」を押して「入力状態」に入ります。入力状態は全画面編集モード (携帯電話仕様) となります。

「入力状態」で「Center (Select) キー」を押すと「入力状態」を抜けます。

<携帯で Center (Select) キーが押されたときの動作>

- ・ \$Click がある場合 実行
- ・ チェックボックス ON
- ・ テキストコントロール 入力状態に入る **注) \$Click があると入力状態には入りません。**

<エミレータの操作> 携帯電話では一部操作が異なります。

- ・ Enter キーで入力状態へ。
- ・ 入力状態でカーソルキー押下すると選択モードへ。
- ・ マウスクリックは直接入力モードへ。

システムのデフォルト移動のキャンセル

EnableKey 関数に設定する値により元のキー機能を維持するか停止するかを指定できます。

<移動キー、Center キー> | <それ以外のキー>

- | | | |
|-------------|--|-----------|
| 1： (機能する) | | (機能する) |
| 2： (機能する) | | × (機能しない) |
| 3：× (機能しない) | | (機能する) |
| 4：× (機能しない) | | × (機能しない) |
- 5： \$Onkey の戻り値によって機能を有効にする (戻り値 0：機能する 1：機能しない)

例) EnableKey(3) 文字入力可、システムによる自動移動キャンセル

- リスト、ドロップダウンリスト選択モードのキャンセル
リストの選択モードから抜けるためのキー (デフォルト: クリアキー) を設定できます。
・キャンセルキー設定: **ChgCancelKey** 関数 **注) 実機でのみ有効となります。**

(2) 動的コントロール

動的コントロールはツールバーを使ってあらかじめフォーム上に定義しておく通常のコントロールとは異なり、処理式 (**CreateArray** 関数) によってダイナミックに生成するコントロールです。動的コントロールではコントロールを配列として使用することができます。配列 (数) 種類を指定してコントロールを生成し、その後、生成したコントロールに対しそれぞれのプロパティを指定します。

使用方法 (手順)

1. **CreatArray** 関数で、配列名、コントロール種別を指定してコントロールを生成します。
2. 処理式の代入によりプロパティ値をセットします。¹
3. **ShowObj**("コントロール名",0);で表示します。²
4. 通常のコントロールと同様に代入・参照ができます。
5. 動的コントロールは、フォームの **UnLoadForm** で自動的に解放 (消滅) します。

通常のコントロールとの相違点・注意点

動的作成 (配列) コントロールは、

1. **Name** 以外の全てのプロパティが変更可能。
2. **ShowObj()**、**SetFocus()** 等表示関連の関数の対象は、**Height** と **Width** が >0 のコントロールに限る。
3. **UnLoadForm** 時に関連付けられた動的作成 (配列) コントロールは削除される。
4. **CreateArray** 関数呼び出し直後のデフォルト値は "**PadPopupFlg**" 以外定義と同じ。
"**PadPopupFlg**" は "**FALSE**".

例)

データ分のテキストコントロールを作成

```
INT CNT;
D_BLOCK("DB001",1,"","");
//動的テキストの定義
CreateArray("A_TXT","TEXT","FORM01",EF_RECC);
FOR CNT = 0 TO (.DB001.MAX_LIN - 1) //件数分のプロパティ表示
.A_TXT [CNT].DataLength = 80;
.A_TXT [CNT].TOP = 20*(1+CNT);
.A_TXT [CNT].HEIGHT = 20;
.A_TXT [CNT].LEFT = 20;
.A_TXT [CNT].WIDTH = 200;
.A_TXT [CNT] = .DB001.NAIYO;
NEXT
```

¹ **.XXXX[0].TOP=10;** の様に配列で代入・参照できます。

² **\$Load** で **Height** と **Width** をセットした配列のコントロールは、**ShowObj()** をしなくても **\$Load** の最後で表示されます。また、タブオーダーも自動で設定されます。表示の対象は、**Height** と **Width** が >0 のコントロール。**ShowObj()** は、"**.XXXX**" で配列全体、"**.XXXX[m]**" で配列の内の指定されたコントロールだけを対象にする事ができる。**\$Load** 以降は、**ShowObj()** を呼ばなければ表示されない。

(3) コントロール名の引数による代入、参照関数の追加

コントロール名に変数を使ってコントロールへの操作 (コントロールに値を代入、または参照) ができます。(プロパティも可)

指定コントロール名の値を取得 : **GetValueByName** 関数

値を指定コントロール名に代入 : **SetValueByName** 関数

```
例) Text01 ~ Text03 までの値を取得し L_1 ~ L_3 のラベルにセット
CHRA WTEXT*6; //値を取得するテキストコントロール名用
CHAR CPARA*20; //値を保存するエリア
CHAR WOUT*4; //値をセットするコントロール名用
$Click.Cmd11
    WTEXT(1,5) = ".Text";
    FOR i = 1 TO 3
        WTEXT(6,1) = i;
        GetValueByName(WTEXT, &CPARA); //値を取得
        WOUT(1,3) = ".L_"; WOUT(4,1) = i;
        SetValueByName(WOUT, CPARA); //値をセット
    NEXT
$End
```

(4) リストコントロール

新規コントロールとしてリストコントロールが追加されました。グリッド同様一覧型ですがデータ表示のみで入力できません。また、項目の表示順はグループ (DB) で指定されている順で表示します。

リストコントロールのプロパティ

・プロパティ :

Name	コントロール名
BackColor	リストの背景色
BorderColor	リストの枠線色
Focus	フォーカスの有無 (0 : 受け取る、1 : 受け取らない)
FontName	フォント名
FontSize	リスト全体のフォントサイズ
FontWeight	フォントの太さ (0 : 通常、1 : 太い)
ForeColor	リストの文字色
GroupName	リンクするグループ名
HeaderTilte	タイトルの表示有無 (1 : 表示、0 : 非表示)
Height	リストの高さ (ドット数)
Width	リストの幅 (ドット数)
Top	リストの左上の Y 座標
Left	リストの左上の X 座標
InActivCBackColor	非アクティブ時のカレント行の背景色
InActivCForeColor	非アクティブ時のカレント行の文字色
ListStatus	リストのステータス (0 : 選択可能、1 : 選択不可、2 : 非表示)
LineNoTitle	行番号 (行番号なし、1 桁、2 桁、3 桁、4 桁、5 桁)
TitleFontName	タイトルのフォント名
TitleFontSize	タイトルのフォントサイズ
TitleFontWeight	タイトルのフォントの太さ

TitleBackColor	タイトルの背景色
TitleForeColor	タイトルの文字色
Vscrollbar	垂直スクロールバーの表示有無 (0:非表示、1:表示)
SyncScroll	連動スクロールの有無 (0:連動しない、1:連動する)
TitleHeight	タイトルの高さ(カラムの定義が <i>イダ</i> で入力)
RowHeight	カラムの高さ(カラムの定義が <i>イダ</i> で入力)
LineAttrItem	行属性指定アイテム(カラムの定義が <i>イダ</i> で入力)
・カラムのプロパティ	
ItemName	リンクしたグループのアイテム名
Caption	カラムのタイトル
Name	カラム名
ColumnWidth	カラムの幅 (桁数)
(Width	カラムの幅 (換算ドット数: ColumnWidth × FontSize)
Type	0:文字型 1:数値型
Mask	マスク (長さは表示長)
Vborder	カラムの縦境界線の有無 (0:なし、1:あり)
Hborder	カラムの横境界線の有無 (0:なし、1:あり)
Alignment	位置指定 (0:左揃え、1:中央揃え、2:右揃え)
FontSize	カラムのフォントサイズ (-1:リストのサイズを使用)
FontWeight	フォントの太さ (-1:リストのサイズを使用、0:通常、1:太い)
FontName	フォント名
BackColor	カラムの背景色
ForeColor	カラムの文字色
TitleFontSize	タイトルのフォントサイズ (-1:リストのサイズを使用)
TitleFontWeight	タイトルのフォントの太さ (-1:リストのサイズを使用、0:通常、1:太い)
TitleFontName	タイトルのフォント名
TitleBackColor	タイトル毎の背景色
TitleForeColor	タイトル毎の文字色
HBordColor	カラムの横境界線色
VBordColor	カラムの縦境界線色

リストコントロールの対応イベント

イベントは \$Click、\$GotFocus、\$LostFocus があります。

```
書式: $Click.<リスト名>({ INT <ArryNo>; } INT <ItemNo>; INT <Row>;)
      $GotFocus.<リスト名>({ INT <ArryNo>; } INT <ItemNo>; INT <Row>;)
      $LostFocus.<リスト名>({ INT <ArryNo>; } INT <ItemNo>; INT <Row>;)
```

<ArryNo> : リストを動的に生成した場合の Array 番号

<ItemNo> : 連結されているグループのアイテム番号 (カラム番号ではありません)

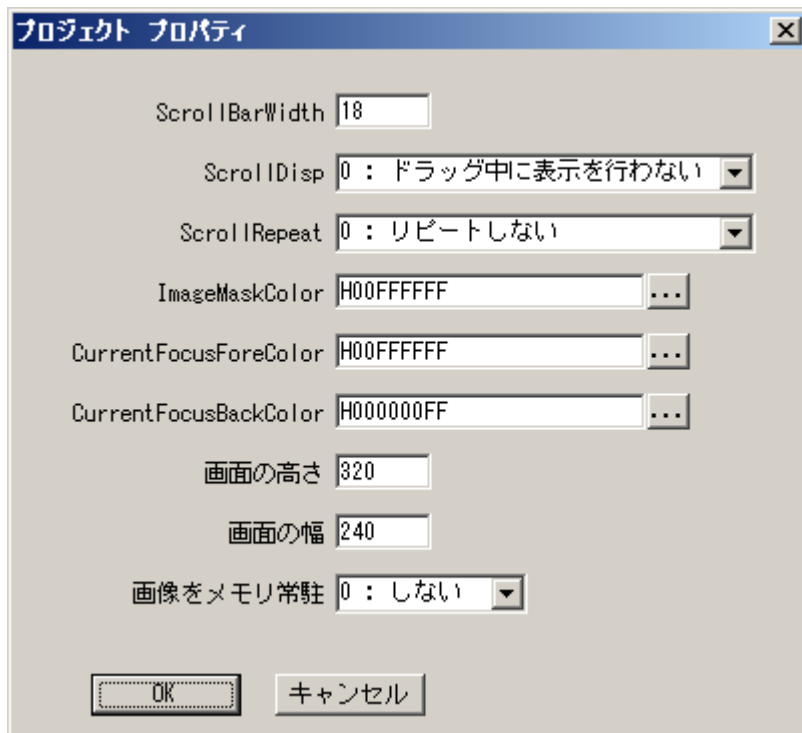
<Row> : 行番号

リストの行 (指定行、または指定行数) スクロール: [ScrollList](#) 関数

リストの表示アイテム切り替え: [ListDispItem](#) 関数

(5) プロジェクトプロパティ

プロジェクトのプロパティが新設されました。プロジェクト共通のプロパティを設定できます。



プロジェクトプロパティ

- ・ **ScrollBarWidth** スクロールバーの幅
- ・ **ScrollDisp** ドラッグ時の表示
(0: ドラッグ中に表示を行わない、1: 行う)
- ・ **ScrollRepeat** リピート (0: リピートしない、1: する)
- ・ **ImageMaskColor** 透過色の指定
- ・ **CurrentFocusForeColor** カレントフォーカスの文字色
- ・ **CurrentFocusBackColor** カレントフォーカスの背景色
- ・ 画面の高さ 新フォーム作成時反映 (デフォルト 320 ドット)
- ・ 画面の幅 新フォーム作成時反映 (デフォルト 240 ドット)
- ・ 画像をメモリに常駐 画像のメモリ常駐指定
(0: 常駐しない、1: 常駐する)
使用頻度の高い画像を常駐させることにより表示のパフォーマンスを上げることができます。ただしメモリ容量を考慮する必要があります。
動的にリソース化した画像はメモリ常駐します。

(6) 画像のリソース化

画像をリソースとして持ち、リソースとして扱うことができるようになります。

リソース化された画像は、コントロールのキャプションやデータの文字列内に表示することができます。また、あらかじめリソース化された画像は、LC ファイル内に内蔵されるため、実行環境では画像ファイルとして個別に持つ必要はありません。

リソース化の方法は、あらかじめプロジェクトのリソース設定で定義しておく方法と、関数によりダイナミックに行う方法があります。

注意) 携帯電話ではリソース化して使用できる画像は BMP のみとなります。(携帯電話の仕様)

プロジェクトでのリソース設定画面



関数によるリソース化

1. リソース領域を確保：RscAlloc 関数
処理式で動的に画像をリソース化します。
2. リソースにプロパティを設定：RscSetProp 関数
RscAlloc 関数、およびあらかじめプロジェクトでリソース定義されたリソースに対してプロパティ (画像名、位置、サイズ、制御コード、画像透過フラグ) を設定します。
3. リソース領域の解放：RscFree 関数

例)

```
RscAlloc ("Ball01"); //リソース Ball01 を作成する
//リソース Ball01 にプロパティを設定
RscSetProp(".Ball01.Picture"," Ball01.BMP"); //画像ファイル名
RscSetProp(".Ball01.ControlCode","¥XFD¥X01"); //制御コード
RscSetProp(".Ball01.Width",15); //サイズ幅
RscSetProp(".Ball01.Height",15); //サイズ高さ
RscSetProp(".Ball01.PosX",20); //位置 X (コントロール内相対位置)
RscSetProp(".Ball01.PosY",3); //位置 Y (コントロール内相対位置)
RscSetProp(".Ball01.ImageMaskFlag",1); //画像透過
```

リソースの表示方法

リソース化された画像の表示方法には次の 3 通りがあります。

- ・コントロールプロパティの Picture に画像名の代わりに「リソース名」を指定。
例) .Image01.Picture = "Ball01"; または .Image01.Picture = ".Ball01";
- ・コントロールのキャプションやデータに制御コードを指定 (データ内に絵文字として表示される)
例) .Text02 = "ABC¥XFD¥X01XYZ";
- ・リソース名を指定して設定。この場合、データ内の表示位置と表示サイズが指定できます。
例) .Text01 = "¥¥\$. Ball01,40,5,20,25¥¥\$ ABC XYZ";

注意) 設定するコントロールの DataLength は設定内容の分大きめに確保しておく必要があります。

(7) 画像表示の拡張 (透過、拡大縮小、絵文字化、動的な表示)

画像の扱いが拡張されました。

透過

- プロジェクトプロパティで **ImageMaskColor** に透過色を指定し、透過させる画像に対しては、イメージプロパティやリソース定義で **画像透過フラグ (ImageMaskFlag)** を ON を指定します。

拡大縮小

- イメージプロパティで **SizeAdjust** (0 以外を指定) を指定することにより、画像を任意のサイズに拡大 / 縮小することができます。

絵文字化

- リソースに割り当てた **制御コード (ControlCode)** を文字列の中に指定すると「絵文字」の様に表示されます。

例) `.Text01 = "aaa¥xFD¥x59bbb¥xFD¥x60";`

リソースによる動的な表示

- 処理式内で **リソース名** を指定することによりサイズ、位置など動的に変更して表示することができます。

"¥¥\$ ~ ¥¥\$" で囲んで、リソース名、プロパティを指定します。プロパティ指定が省略された場合は、SrcSetProp 関数でセットしたプロパティが使用されます。(SrcSetProp 関数で位置、サイズのプロパティが設定されていない場合は、位置 0,0、元画像サイズとなります)

注意) 設定するコントロールの DataLength は設定内容の分大きめに確保しておく必要があります。

例) `.Text01 = "¥¥$.AAAA,40,5,20,25¥¥$ ABC XYZ";`

解説: `.Text01` にリソース名「AAAA」を `.Text01` の左上を基準に `x=40,y=5 width=20,height=25` で表示します。

(8) 現在および一つ前のフォーカスのコントロール名取得

現在 (カレント) フォーカスがあるコントロール情報取得: **CurFocus** 関数

一つ前のフォーカスがあったコントロー情報取得: **PrvFocus** 関数

注意) 取得したコントロール名の判別時は大文字、小文字を意識して指定してください。

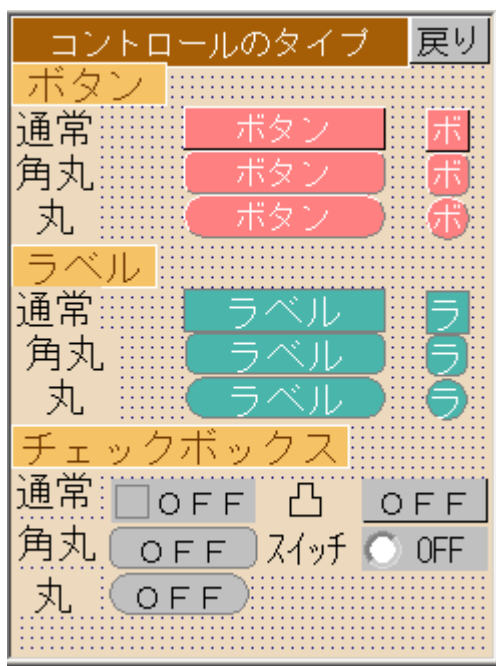
(9) ButtonAction の「CALL」の拡張

「ButtonAction」「CALL」の「ButtonActPrm」に本体のキーコード (\$OnKey で取得するキー値) を設定することにより、ボタンクリックでキーイベントを発行できます。

例) 35 エンターキー (PC の場合の例)
25 上矢印キー (PC の場合の例)

(10) コントロールの形状の拡張 (角丸、端丸等)

ボタン、ラベル、チェックボックスのプロパティにスタイルが追加され、角丸・端丸が設定できます。



ボタンコントロール

プロパティの **Style** に設定します。

0 : 四角 1 : 角丸 2 : 端丸

1 の時、「高さ / 2 > 幅」の場合、四角になります。

2 の時、「高さ > 幅」の場合、四角になります。

処理式で動的に設定する場合

例) `.Btn01.Style = 2;`

ラベル、チェックボックス

プロパティの **BorderStyle** に設定します。

5 : 角丸 6 : 端丸

5 の時、「高さ / 2 > 幅」の場合、四角になります。

6 の時、「高さ > 幅」の場合、四角になります。

処理式で動的に設定する場合

例) `.Label01.BorderStyle = 6;`

`.Check01.BorderStyle = 6;`

角丸、端丸の場合、**BorderColor** は指定できません。

(1 1) ドロップダウンリストの拡張 (表示行数、フォントの指定が可能)

ドロップダウンリストのプロパティに表示フォント **ListFontSize**、表示行数 **ListLines** が追加されました。

- ・ **ListLines** (0 : 自動 (デフォルト))
- ・ **ListFontSize** (0 : コントロールにあわせる (デフォルト))

(1 2) スクロールバーの拡張 (幅、ドラッグ動作、リピートの指定が可能)

スクロールバーの幅、ドラッグ時の動き、リピートの有無が指定できます。

プロジェクトプロパティにて指定

- ・ **ScrollBarWidth** スクロールバーの幅
- ・ **ScrollDisp** ドラッグ時の表示
(0 : ドラッグ中に表示を行わない、 1 : 行う)
- ・ **ScrollRepeat** リピート (0 : リピートしない 1 : する)

(1 3) フォントの拡張 (ボールド)

プロパティにフォントの太さ **FontWeight** が追加されました。

- ・ **FontWeight** (0 : 通常、 1 : 太い) 注)機種依存

(1 4) 入力方式の拡張 (上書きモード、電卓入力、文字チェック、数値の桁数指定)

テキストコントロールの入力関係のプロパティに上書き **SelectAll**、電卓入力 **Calculator**、文字チェック **CharCheck**、数値の桁数 **NumericMaxSize** が追加されました。

- ・ **SelectAll** (0 : なし、 1 : 初期全選択状態)
- ・ **Calculator** (0 : 通常、 1 : 電卓入力モード) (数値の場合のみ)
- ・ **CharCheck** (0 : すべて可、 1 : 数字可、 2 : 英大文字可、 4 : 英小文字可、
8 : 記号可、 1 6 : 半角カナ可、 3 2 : 全角可)
- ・ **NumericMaxSize** (0 : 無制約 n : 桁数) (数値の場合のみ)

(1 5) テキストコントロールの拡張 (MASK、Alignment)

テキストコントロールのプロパティに **Mask**、**Alignment** が追加されました。

(1 6) フォーム背景への画像貼付け

フォームのプロパティに背景画像 **BGPicture** が追加されました。BGPicture には画像ファイル、およびリソース名が指定できます。

(1 7) イベント追加 (イメージコントロールに\$Click イベント)

イメージコントロールに\$Click イベントが追加されました。

(1 8) カレントフォーカスの色指定

カレントフォーカスの色指定が追加されました。

プロジェクトプロパティで設定

- ・ **CurrentFocusForeColor** カレントフォーカスの文字色
- ・ **CurrentFocusBackColor** カレントフォーカスの背景色

動的にカレントフォーカスの色を設定：[SetCrtCrsClr](#) 関数

3.関数リファレンス

3.1. フォーム関連関数 (共通)

■ CreateArray (クエイト・アレイ) コントロールの生成

機能 動的にコントロールを生成します

書式 <戻り値> = CreateArray(<コントロール名> , <種類> , <フォーム名> , <生成数>)

設定項目	内容	型
<コントロール名>	生成するコントロール名	整数型
<種類>	コントロール種別 "TEXT"、"BUTTON"、"LABEL"、"CHECK"、"LIST"	文字型
<フォーム名>	対象フォーム名 生成するコントロールが関連付けられるフォームを指定します。(ロードされているフォームでなければなりません)	文字型
<生成数>	生成するコントロールの数 (配列数)	整数型
<戻り値>	0 : 成功 - 1 : 失敗	整数型

解説 指定された数のコントロールを配列として生成します。同名 Form 名同名 Array 名での再定義はエラーとなります。本関数で生成したコントロールは該当フォームを UnLoadForm した時点で全て削除されます。

本関数で生成したコントロールに対しては、Name 以外の全てのプロパティに対し動的な変更可能となります。生成時のデフォルトプロパティは手動定義時と同様の初期値がセットされます。

注意) 本関数で生成するコントロールは実行時に動的に生成され表示しますので、他のコントロールと重ならないよう注意が必要です。

使用方法 (手順)

1. CreatArray 関数で、配列名、コントロール種別を指定してコントロールを生成します。
2. 処理式の代入によりプロパティ値をセットします。³
3. ShowObj("コントロール名",0);で表示します。⁴
4. 通常のコントロールと同様に代入・参照ができます。
5. 動的コントロールは、フォームの UnLoadForm で自動的に解放 (消滅) します。

処理式で配列を使用する場合、添字は[0] ~ [<生成数> -1]となります。

書式: .コントロール名[添字]

³ .XXXX[0].TOP=10; の様に配列で代入・参照できます。

⁴ \$Load で Height と Width をセットした配列のコントロールは、ShowObj()をしなくても \$Load の最後で表示されます。また、タブオーダーも自動で設定されます。表示の対象は、Height と Width が>0 のコントロール。ShowObj()は、".XXXX"で配列全体、".XXXX[m]"で配列の内の指定されたコントロールだけを対象にする事ができます。\$Load 以降は ShowObj()を呼ばなければ表示されません。

サンプル 1) データ分のテキストコントロールを作成

```
INT CNT;
D_BLOCK("DB001",1,"",""); //データ読み込み
//動的テキストコントロールの定義
CreateArray("A_TXT","TEXT","FORM01",EF_RECC);
FOR CNT = 0 TO (.DB001.MAX_LIN - 1) //件数分のプロパティ設定
  .A_TXT [CNT].DataLength = 80;
  .A_TXT [CNT].TOP = 20*(1+CNT);
  .A_TXT [CNT].HEIGHT = 20;
  .A_TXT [CNT].LEFT = 20;
  .A_TXT [CNT].WIDTH = 200;
  .A_TXT [CNT] = .DB001.NAIYO;
NEXT
```

■ CurFocus (カレント・フォーカス) 現在のフォーカス情報取得

機能 カレントのフォーカス情報を取得します

書式 <戻り値> = CurFocus (&<オブジェクト> { , &<COL> , &<ROW> , &<値> })

設定項目	内容	型
<オブジェクト>	取得した現在のフォームまたはコントロール名 書式："フォーム名"、または"コントロール名" 動的生成コントロールの場合は、"Array 名[n]"の形式となります。	文字型
<COL>	取得した現在のカラム番号	数値型
<ROW>	取得した現在の行番号	数値型
<値>	取得した現在のコントロールの値	コントロールによる
<戻り値>	- 1 : エラー それ以外 : 正常	数値型

解説 現在 (カレント) フォーカスがあるコントロールの内容を取得します。

注意) 取得したオブジェクト名は大文字小文字を区別しますので、IF 文等で判断する際は定義した名前で記述するようにして下さい。

サンプル

■ PrevFocus (プレビウス・フォーカス) 一つ前のフォーカス情報取得

機能 一つ前のフォーカス情報を取得します

書式 <戻り値> = PrevFocus (&<オブジェクト> { , &<COL> , &<ROW> , &<値> })

設定項目	内容	型
<オブジェクト>	取得した一つ前のフォームまたはコントロール名がセットされます。 書式：".フォーム名"、または".フォーム名.コントロール名" 動的生成コントロールの場合は、".FORM 名.Array 名[n]" の形式となります。	文字型
<COL>	取得したカラム番号	数値型
<ROW>	取得した行番号	数値型
<値>	取得したコントロールの値	コントロールによる
<戻り値>	- 1 : エラー それ以外 : 正常	数値型

解説 一つ前にフォーカスのあったコントロールの内容を取得します。

注意) 取得したオブジェクト名は大文字小文字を区別しますので、IF 文等で判断する際は定義した名前で記述するようにして下さい。

サンプル

■ GetValueByName (ゲットバリュー・バイ・ネーム) 名前によるコントロール値取得

機能 指定コントロール名の値を取得します

書式 <戻り値> = GetValueByName (<コントロール名> , &<値>)

設定項目	内容	型
<コントロール名>	値を取得したいコントロール名 (プロパティ名) 書式: ".コントロール名" または ".コントロール.プロパティ名"	文字型
<値>	取得したコントロールの値 (プロパティ値)	コントロールによる
<戻り値>	- 1 : エラー それ以外: 正常	数値型

解説 コントロール名を引数として指定してその値を取得します。コントロール名.プロパティ名を指定した場合はプロパティ値を取得します。
コントロール名を変数で渡すことにより、複数のコントロールの参照や代入の記述を変数を使って共通化できます。

サンプル 例) Text01 ~ Text03 までの値を取得し Label01 ~ Label03 のラベルにセット

```
CHRA WTEXT*6; //値を取得するテキストコントロール名用
CHAR CPARA*20; //値を保存するエリア
CHAR WOUT*7; //値をセットするコントロール名用
$Click.Cmd11
    WTEXT(1,5) = ".Text";
    FOR i = 1 TO 3
        WTEXT(6,2) = i;
        GetValueByName(WTEXT, &CPARA); //TextXX から値を取得
        WOUT(1,5) = ".Label";
        WOUT(6,2) = i;
        SetValueByName(WOUT, CPARA); //LabelXX に値をセット
    NEXT
$End
```

■ SetValueByName (セットバリュ・バイ・ネーム) 名前によるコントロール値設定

機能 指定コントロール名へ値を設定します

書式 <戻り値> = SetValueByName (<コントロール名> , <値>)

設定項目	内容	型
<コントロール名>	値を設定するコントロール名 (プロパティ名) 書式: ".コントロール名" または ".コントロール.プロパティ名"	文字型
<値>	設定するコントロールの値 (プロパティ値)	コントロールによる
<戻り値>	- 1 : エラー それ以外: 正常	数値型

解説 コントロール名を引数として指定してそのコントロールに値を設定します。コントロール名.プロパティ名を指定した場合はプロパティ値を設定します。コントロール名を変数で渡すことにより、複数のコントロールへの代入記述を変数を使って共通化できます。

サンプル 例) Text01 ~ Text03 までの値を取得し Label01 ~ Label03 のラベルにセット

```
CHRA WTEXT*6; //値を取得するテキストコントロール名用
CHAR CPARA*20; //値を保存するエリア
CHAR WOUT*7; //値をセットするコントロール名用
$Click.Cmd11
    WTEXT(1,5) = ".Text";
    FOR i = 1 TO 3
        WTEXT(6,2) = i;
        GetValueByName(WTEXT, &CPARA); //TextXX から値を取得
        WOUT(1,5) = ".Label";
        WOUT(6,2) = i;
        SetValueByName(WOUT, CPARA); //LabelXX に値をセット
    NEXT
$End
```

■ RscAlloc (リソース・アロケート) リソース領域の確保

機能 リソース領域を確保します

書式 <戻り値> = RscAlloc (<リソース名>)

設定項目	内容	型
<リソース名>	確保するリソース名を指定	文字型
<戻り値>	- 1 : エラー 0 : 正常	数値型

解説 リソース領域を確保します。
リソース化された画像は、コントロールのキャプションやデータの文字列内に表示することができます。また、あらかじめリソース化された画像は.LC ファイル内に内蔵されるため、実行環境では画像ファイルとして個別に持つ必要はありません。
リソース化の方法は、あらかじめプロジェクトのリソース設定で定義しておく方法と、本関数によりダイナミックに行う方法があります。
注意) 携帯電話ではリソース化して使用できる画像は BMP のみとなります。(携帯電話の仕様)

サンプル 例)

```
RscAlloc ("Ball01"); //リソース Ball01 を作成する
//リソース Ball01 にプロパティを設定
RscSetProp(".Ball01.Picture"," Ball01.BMP"); //画像ファイル名
RscSetProp(".Ball01.ControlCode","¥XFD¥X01"); //制御コード
RscSetProp(".Ball01.Width",15); //サイズ幅
RscSetProp(".Ball01.Height",15); //サイズ高さ
RscSetProp(".Ball01.PosX",20); //位置 X
RscSetProp(".Ball01.PosY",3); //位置 Y
RscSetProp(".Ball01.ImageMaskFlag",1); //画像透過
```

■ RscSetProp (リソース・セット・プロパティ) リソースプロパティの設定

機能 リソースにプロパティを設定します

書式 <戻り値> = RscSetProp (<リソースプロパティ名> , <値>)

設定項目	内容	型
<リソースプロパティ名>	リソース名+プロパティ名を指定 書式: ".リソース名.プロパティ名"	文字型
<値>	プロパティの値を指定	プロパティによる
<戻り値>	- 1 : エラー 0 : 正常	数値型

解説 リソースにプロパティを設定します。
RscAlloc 関数、およびあらかじめプロジェクトでリソース定義されたリソースに対してプロパティ (画像名、位置、サイズ、制御コード、画像透過フラグ) を設定します。

サンプル 例)

```
RscAlloc ("Ball01"); //リソース Ball01 を作成する
//リソース Ball01 にプロパティを設定
RscSetProp(".Ball01.Picture", "Ball01.BMP"); //画像ファイル名
RscSetProp(".Ball01.ControlCode", "¥XFD¥X01"); //制御コード
RscSetProp(".Ball01.Width", 15); //サイズ幅
RscSetProp(".Ball01.Height", 15); //サイズ高さ
RscSetProp(".Ball01.PosX", 20); //位置 X
RscSetProp(".Ball01.PosY", 3); //位置 Y
RscSetProp(".Ball01.ImageMaskFlag", 1); //画像透過
```

■ RscFree (リソース・フリー) リソース領域の開放

機能 リソース領域を開放します

書式 <戻り値> = RscFree ()

設定項目	内容	型
<戻り値>	- 1 : エラー 0 : 正常	数値型

解説 リソース領域を開放します。
なお、確保されたリソースはアプリケーション終了時に開放されます。

■ SetCrtCrsClr (セツ・カレントカーソル・カラー) カレントフォーカス色の設定

機能 カレントフォーカスの色を設定します

書式 <戻り値> = SetCrtCrsClr (<背景色> , <文字色>)

設定項目	内容	型
<背景色>	カレントフォーカスの背景色 (CurrentFocusBackColor) 書式 : 0x00000000	数値型
<文字色>	カレントフォーカスの文字色 (CurrentFocusForeColor) 書式 : 0x00000000	数値型
<戻り値>	- 1 : エラー 0 : 正常	数値型

解説 カレントフォーカスの色を動的に設定 / 変更します。

カレントフォーカスの色は、予めプロジェクト・プロパティで設定しておくことができます。

サンプル 例) 背景 : 赤、文字色 : 白の場合
SetCrtCrsClr(0x000000FF,0x00FFFFFF);

■ ScrollList (スクロール・リスト) リスト行スクロール

機能 リストの行スクロール

書式 <戻り値> = ScrollList (<リスト名> , <行> { , <モード> })

設定項目	内容	型
<リスト名>	リストコントロール名 書式: ".フォーム名.コントロール名"	文字型
<行>	リスト内の行番号、またはスクロールする行数 <モード> により	数値型
<モード>	スクロールのモード 0 : 絶対 指定行までスクロール (規定値) 1 : 相対 現在行から行数分スクロール	数値型
<戻り値>	- 1 : エラー 0 : 正常	数値型

解説 リストコントロールの行 (縦方向) に対するスクロールを行います。

サンプル 例) 10行分スクロールする場合
ScrollList(".Form01.List01", 10, 1);

■ ListDispItem (リスト・ディスプレイ・アイテム) リスト表示アイテム切り替え

機能 リストの表示アイテム切り替え

書式 <戻り値> = ListDispItem (<リスト名> , <アイテム名>)

設定項目	内容	型
<リスト名>	リストコントロール名 書式: ".フォーム名.コントロール名"	文字型
<アイテム名>	表示するアイテム名 カンマで区切って複数指定できます。 書式: "アイテム名1, アイテム名2 . . ."	文字型
<戻り値>	- 1 : エラー 0 : 正常	数値型

解説 リストコントロールに表示するアイテムを切り替えます。<アイテム名>に指定したアイテムの順に左から表示されます。
本関数実行時にコントロールの再描画が行われます。

サンプル 例) アイテムF4、F2をF4、F2の順に表示する場合
llistDispItem(".Form01.List01", "F4,F2");

3.2. 携帯電話専用関数 (携帯共通)

■ SetControlTab フォーカス移動順序設定

機能 フォーカスの移動モードを設定します

書式 LOCALFUNC("SetControlTab", <スイッチ>, <フォーム名> {, <モード> })

設定項目	内容	型
<スイッチ>	1 : 上 下 左 右順 (規定値) 2 : 左 右 上 下順	整数型
<フォーム名>	対象フォーム名	文字型
<モード>	0 : ぐるぐる回る (規定値) 1 : 末端で止まる	整数型

解説 TabOrder の設定がある場合は該当テキストボックスのみその順番で移動します。TabOrder の設定が無い場合はフォーカス対象コントロール全て<モード>で指定した順で移動します。(いずれのコントロールにもフォーカスが無い場合は無効です) 対象のコントロールは、テキストボックス、コマンドボタン、チェックボックスです。
(注意)

- ・ 本関数は動的作成されたコントロールも含めて設定されます。動的作成コントロールの TOP・LEFT のデフォルト値は 0 であるため、TOP・LEFT 座標を設定してから関数を呼び出さないと正しく TabOrder が設定されません。
- ・ \$OnKey ブロックにて独自に キーを判定して同一フォーム内の任意のコントロールに Setfocus している場合は、その移動制御と TabOrder の移動制御が重なるため、Return (1);を行い、システムによる自動移動をキャンセルする必要があります。(下記サンプル 1 参照)

サンプル LocalFunc("SetControlTab",2,"FORM01");

```

1
SONKEY(INT KEY;)
  SWITCH(KEY)
  CASE (AVK_UP) /* */
    IF iSWX == 0 THEN
      SetFocus(".FM01.LIST1[0]",1,1);
      iSWX = 1;
      RETURN (1); // TabOrder による自動移動キャンセル
    ENDIF
  BREAK;

```

■ CursorKeyMode (カーソルキーモード) 移動キー設定

機能 フォーカスの移動キーを設定します

書式 LOCALFUNC("CursorKeyMode", <モード>, <フォーム名>)

設定項目	内容	型
<モード>	0 : キーで移動 (規定値) 1 : キーで移動 2 : キーで上下左右に移動 左右に近いコントロール。上下の近いコントロールに移動 (左上を起点)	整数型
<フォーム名>	対象フォーム名	文字型

解説 フォーカスの移動キーを設定します。
フォーム単位で指定します。

サンプル LocalFunc("CursorKeyMode",1,"FORM01");

■ ChgCancelKey (チェンジ・キャンセルキー) キャンセルキー変更

機能 選択モードから抜けるキーを設定します

書式 LOCALFUNC("ChgCancelKey", <キー>)

設定項目	内容	型
<キー>	キーコード 注意) 携帯電話機種により異なります。	整数型

解説 リストコントロール、およびドロップダウンリストの選択モードから抜ける(キャンセル)するためのキー(デフォルトでクリアキーが割り当てられています)を設定します。
(エミュレータでは[ESC]キーが割り当てられています)

サンプル LocalFunc("ChgCancelKey", 0xE036); //Soft キーをキャンセルキーにセット

3.3. 携帯電話専用関数 (機種専用)

機種デPENDの関数については、対応機種の検証用キットに付属の「専用 API 説明書_BREW.doc」をご参照ください。