

COMWARE

帳票作成/印刷支援ツール **COREPO**
DLL版 印刷エンジン
リファレンス ガイド

ご注意

- ・本製品の一部または全部を弊社の書面による許可なく複写・複製することは、その形態を問わず、禁じます。
- ・このソフトウェアは、コンピュータ1台につき1セット購入が原則になっております。
- ・本製品の内容・仕様は、訂正・改善のため予告なく変更することがあります。
- ・本製品に同封のソフトウェア使用許諾書に基づいて個人で使用する場合を除いて、弊社の承認なしにこのソフトウェアおよびマニュアルを使用することを固くお断りします。
- ・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責任を負いかねますので、ご了承ください。
- ・本製品の内容については万全を期しておりますが、万一ご不審な点やお気づきの点がございましたら、弊社まで連絡ください。

Microsoft、Windows、Visual Basic、Visual C++、Visual Studio、Visual Studio.Net は米国マイクロソフト社の登録商標です。
C++Builder、Delphi は米国ボウランド社の登録商標です。

帳票作成・印刷支援ツール

C O R E P O

Copyright(C)2003 有限会社コムウェア
All rights reserved.

本マニュアルの一部、または全部を、有限会社コムウェアの
書面による許可なく複製、送信、複写、転載、情報検索のための保存、
および他の言語に翻訳することを禁止します。

開発：有限会社コムウェア
〒373-0801 太田市台之郷町 601-2
TEL 0276 48 0811
FAX 0276 48 0815

目次

ごあいさつ	7
サポート	8
動作環境	9
特徴	10
第1章 イン트로ダクション	11
1.1 レポート印刷エンジンのリンク方法	11
1.2 配布可能ファイル	13
1.3 プログラミングの概要	14
1.4 印刷エンジン関数の一覧	15
第2章 関数リファレンス	21
2.1 印刷エンジンセッションの関数	21
cwOpenEngine	21
cwCloseEngine	22
cwGetVersion	23
cwSetErrorMode	24
cwGetLastError	25
cwPrinterSetupDlg	26
cwSetInternetParam	27
cwOpenReport	29
cwOpenPrintFile	30
cwCloseReport	31
cwSelectFmtPage	32
cwSetOffset	33
2.2 レポートアクセス・セッションの関数	34
2.2.1 基本的な関数	34
cwSetText	34
cwGetText	36
cwSetIntToStr	37
cwSetUIntToStr	39
cwSetLongToStr	41
cwSetULongToStr	43
cwSetDoubleToStr	45
cwSetTextByGrpIdx	47
cwGetTextByGrpIdx	49
cwSetIntToStrByGrpIdx	51
cwSetUIntToStrByGrpIdx	53

cwSetLongToStrByGrpIdx	55
cwSetULongToStrByGrpIdx.....	57
cwSetDoubleToStrByGrpIdx.....	59
2.2.2 用紙・プリンタ名アクセス関数.....	61
cwSetPageDef	61
cwGetPageDef	62
cwSetPrinterName.....	63
cwGetPrinterName.....	64
cwSetDEVMODE	65
2.2.3 アイテムプロパティ操作関数.....	66
cwSetImagePath.....	66
cwSetImageDIB.....	68
cwSetDIBits.....	70
cwSetDPI	72
cwClearItemData	73
cwClearAllText	74
cwClearAllBarcode.....	75
cwClearAllImage	76
cwSetBound.....	77
cwGetBound.....	79
cwSetPen	81
cwGetPen	83
cwSetBrush.....	85
cwGetBrush.....	87
cwSetFont.....	89
cwGetFont.....	91
cwSetMultiLine.....	93
cwGetMultiLine.....	94
cwSetTextAlign.....	95
cwGetTextAlign.....	97
cwSetSpace.....	99
cwGetSpace.....	101
cwSetTextEscapement.....	103
cwGetTextEscapement.....	104
cwGetTextMetrics.....	106
cwSetWordwrap.....	108
cwGetWordwrap.....	110
cwSetBarcodeProp.....	112
cwGetBarcodeProp.....	113
cwSetImgStretch	114
cwGetImgStretch	115
cwSetVisible	116
cwGetVisible	117
cwSetGrpVisible	118
cwSetForeground	120
cwGetItemType.....	121
cwRenameItem	122
cwSetFixed.....	123
cwGetFixed.....	124

2.2.4 アイテムリピートモード関数	125
cwStartRepeat	125
cwDupBoxLin	127
cwDupGrp	128
cwCheckDupBound	129
2.2.5 アイテム一覧取得関数	131
cwGetItemList	131
cwGetFirstItem	133
cwGetNextItem	135
cwCloseItemHandle	137
2.2.6 インデックス付きグループのアイテムプロパティ操作関数	138
cwSetImagePathByGrpIdx	138
cwSetImageDIBByGrpIdx	140
cwSetDIBitsByGrpIdx	142
cwSetDPIByGrpIdx	144
cwClearItemDataByGrpIdx	146
cwSetBoundByGrpIdx	148
cwGetBoundByGrpIdx	150
cwSetPenByGrpIdx	152
cwGetPenByGrpIdx	154
cwSetBrushByGrpIdx	156
cwGetBrushByGrpIdx	158
cwSetFontByGrpIdx	160
cwGetFontByGrpIdx	162
cwSetTextAlignByGrpIdx	164
cwGetTextAlignByGrpIdx	166
cwSetSpaceByGrpIdx	168
cwGetSpaceByGrpIdx	170
cwSetTextEscapementByGrpIdx	172
cwGetTextEscapementByGrpIdx	174
cwGetTextMetricsByGrpIdx	176
cwSetWordwrapByGrpIdx	178
cwGetWordwrapByGrpIdx	180
cwSetBarcodePropByGrpIdx	182
cwGetBarcodePropByGrpIdx	184
cwSetImgStretchByGrpIdx	186
cwGetImgStretchByGrpIdx	188
cwSetVisibleByGrpIdx	190
cwGetVisibleByGrpIdx	192
cwSetForegroundByGrpIdx	194
cwSetFixedByGrpIdx	196
cwGetFixedByGrpIdx	198
2.2.7 レポートの生成 / アイテムの生成・削除 関数	200
cwCreateReport	200
cwCreateItem	202
cwDeleteItem	204
2.3 印刷ジョブ・セッションの関数	205
cwOpenPrintJob	205
cwOpenPrintJobEx	207

cwClosePrintJob	209
cwGetPrvWnd	210
cwShowPreView.....	211
cwPrintReport.....	212
cwPrvWndStatus.....	213
cwSetPrvWndBounds.....	214
cwGetPrvWndBounds.....	215
cwClosePrvWnd.....	216
cwSetPrvWndPage	217
cwGetPrvWndPage	218
cwSetPrvWndScale.....	219
cwGetPrvWndScale.....	220
cwSetPrvWndBarStatus.....	221
cwGetPrvWndBarStatus.....	222
cwPrvWndDoPrint	223
2 . 4 データブロック(構造体)	224
2 . 4 . 1 ページ・プロパティ・データブロック(構造体)型	224
CW_PAGEDEF.....	224
CW_BARCODEDEF.....	226
第3章 プログラミング・ガイド	228
3 . 1 印刷エンジン・セッションの開始と終了	228
3 . 2 レポートアクセス・セッションの開始と終了	230
3 . 3 印刷ジョブ・セッションの開始と終了.....	233
3 . 4 リピートモードでの印刷.....	235
3 . 5 印刷データのファイル出力.....	238
3 . 6 印刷データファイルの印刷.....	241
3 . 7 エラーモードの設定.....	245
3 . 8 エラーコードの取得.....	246
3 . 9 用紙やプリンタの設定	248
3 . 9 . 1 印刷時に印刷ダイアログ画面を表示して、用紙やプリンタの設定を行う... 248	
3 . 9 . 2 印刷時、レポート印刷エンジンに直接指定する	248
3 . 10 文字列をテキストアイテムに設定する	253
3 . 11 数値を文字列に変換してテキストアイテムに設定する	255
3 . 12 文字列をバーコードアイテムに設定する	262
3 . 12 . 1 文字列をバーコードアイテムに設定する	262
3 . 12 . 2 CODE 128に関する補足説明.....	263
3 . 12 . 3 EAN 128に関する補足説明.....	265
3 . 12 . 4 CODE 39フルアスキーに関する補足説明.....	266
3 . 13 フォントの色・体裁をテキストアイテムに設定する	268
3 . 14 表示・非表示をアイテムに設定する.....	271
3 . 15 枠や線の色・模様, 塗りつぶしの色・模様をアイテムに設定する	273
3 . 16 イメージをイメージアイテムに設定する	277
付録.....	279
エラーコード.....	279

ごあいさつ

この度は、弊社ソフト「COREPO」をご購入いただきまして、誠にありがとうございます。

「COREPO」(以下「コレポ」と表記)は、帳票をデザインするために必要な様々な機能を取り揃えたもっともパワフルな帳票作成・印刷支援ツールです。

ご使用いただく前に、パッケージに添付されているソフトウェア使用許諾契約書、本リファレンスガイドおよび、ヘルプをお読みいただき、コレポの機能をご活用ください。

なお、製品の内容には万全を期しておりますが、万一ご不審な点、誤り、記載漏れなどお気付きの点がありましたら、お手数ですが弊社までご連絡いただければ幸いです。

有限会社コムウェア
コレポ開発担当者一同

サポート

本製品のサポートは基本的には E-Mail、Fax にて行わせていただいております。また、Web サーバーでの FAQ 掲示、ご意見・ご要望に対処したモジュールを自由にダウンロードできる環境もご用意しておりますので、まず、そちらをご参照ください。

有限会社 コムウェア

〒373-0801 群馬県太田市台之郷町601番地の2

E-Mail crp-support@com-ware.jp

コレポ サポート窓口 TEL 0276-46-2502

FAX 0276-48-0815

電話がつながりにくい場合が発生し、お客様にご迷惑をおかけしています。お手数でも、なるべく、電子メールで、お問合せくださいますようお願い申し上げます。

動作環境

コレポ D L L 版印刷エンジンをご使用いただくには、次の動作環境が必要です。

日本語オペレーティングシステム

Microsoft Windows 95osr2/98se/Me/NT4.0sp3/2000/Xp

コンピュータ本体

- ・ Pentium166MHz 以上のプロセッサを搭載したパーソナルコンピュータ
- ・ 100MB 以上の空き容量を持つハードディスク
- ・ CD-ROM ドライブ
- ・ 日本語オペレーティングシステムが推奨するメモリ容量

特 徴

コレポDLL版印刷エンジンを使えば、C,C++,Visual Basic,Delphi といった DLL にアクセスできるプログラミング言語を使う開発者は、簡単なコーディングで、洗練されたレポートの出力機能をアプリケーションに追加することができます。

コレポDLL版印刷エンジンは、他のアプリケーションに、コレポ・デザイナーで作成したレポートの出力制御の機能を提供するダイナミックリンクライブラリ(DLL)です。コレポ正規ユーザであれば、ロイヤリティなしに、作成したアプリケーションに印刷エンジン DLL (CWREPENG.DLL と CWBARCODE.DLL) を付けて出荷することができます。

第 1 章

イントロダクション

この章は、コレポDLL版印刷エンジンを紹介し、コレポDLL版印刷エンジンおよび、本マニュアルの基本的な使い方に慣れていただくためのものです。

1.1 レポート印刷エンジンのリンク方法

印刷エンジンにアクセスするためには、いくつかのファイル(ライブラリ・インクルードファイル等)のほか、アプリケーションに CWREPENG.DLL を含めなければいけません。必要なランタイムファイルについて詳しくは「配布可能ファイル」を参照してください。

使用する開発言語が違くと、ライブラリファイルの指定方法も違ってきます。以下に示すのは、一般的な言語や開発ツールにおけるコレポ印刷エンジンライブラリのリンク方法です。

Borland C++Builder

Borland C++Builder で開発するアプリケーションからコレポ印刷エンジンへの**動的リンク**を作るには、Windows API の LoadLibrary 関数を使ってください。DLL を使い終わったら、FreeLibrary 関数で CWREPENG.DLL が使ったメモリを開放してください。これらの関数の使い方については、Microsoft Windows SDK を参照してください。

Borland C++Builder で開発するアプリケーションからコレポ印刷エンジンへの**静的リンク**を作るには、CWREPENG.DLL のインポートライブラリを、アプリケーションのプロジェクトファイルに追加します。CWREPENG.DLL のインポートライブラリ(CWREPENG.LIB)は、コレポと同じディレクトリの下の Borland C++\lib (デフォルト C:\Program Files\COREPO\Borland C++) に有ります。

また、CWREPENG.DLL の関数、型や定数は、インクルードファイル(CWREPENG.H)に宣言されています。CWREPENG.DLL の関数を呼び出すアプリケーションのソースプログラムに CWREPENG.H をインクルードしてください。インクルードファイル(CWREPENG.H)は、コレポと同じディレクトリの下 Borland C++\include(デフォルト C:\Program Files\COREPO\Borland C++) に有ります。

Visual C++

Visual C++で開発するアプリケーションからコレポ印刷エンジンへの**動的リンク**を作るには、Windows API の LoadLibrary 関数を使ってください。DLL を使い終わったら、FreeLibrary 関数で CWREPENG.DLL が使ったメモリを開放してください。これらの関数の使い方については、Microsoft Windows SDK を参照してください。

Visual C++で開発するアプリケーションからコレポ印刷エンジンへの**静的リンク**を作るには、CWREPENG.DLL のエクスポートファイルを、アプリケーションのプロジェクトファイルに追加します。CWREPENG.DLL のエクスポートファイル (CWREPENG.EXP)は、コレポと同じディレクトリの下の VC++¥lib (デフォルト C:¥Program Files¥COREPO¥VC++¥lib) に有ります。

また、CWREPENG.DLL の関数、型や定数は、インクルードファイル (CWREPENG.H) に宣言されています。CWREPENG.DLL の関数を呼び出すアプリケーションのソースプログラムに CWREPENG.H をインクルードしてください。インクルードファイル (CWREPENG.H)は、コレポと同じディレクトリの下で VC++¥include (デフォルト C:¥Program Files¥COREPO¥VC++¥include) に有ります。

Visual Basic

Visual Basic で開発するアプリケーションからコレポ印刷エンジンへのリンクを作るには、CWREPENG.DLL の関数宣言ファイル (CWREPENG.BAS) を、アプリケーションのプロジェクトファイルに追加します。関数宣言ファイル (CWREPENG.BAS)は、コレポと同じディレクトリの下で VB Module (デフォルト C:¥Program Files¥COREPO¥VB Module) に有ります。また、必要な関数のみを宣言したい場合は、本ヘルプの関数リファレンスを参考に、開発者自身で宣言してください。

Borland Delphi

Borland Delphi で開発するアプリケーションからコレポ印刷エンジンへのリンクを作るには、CWREPENG.DLL の関数宣言ファイル (CWREPENG.PAS) を、アプリケーションのソースファイルにインクルードします。関数宣言ファイル (CWREPENG.PAS) は、コレポと同じディレクトリの下で Delphi (デフォルト C:¥Program Files¥COREPO¥Delphi) に有ります。また、必要な関数のみを宣言したい場合は、本ヘルプの関数リファレンスを参考に、開発者自身で宣言してください。

その他の開発言語

上記以外の開発言語で使用する場合には、申し訳ございませんが、開発者自身で必要な宣言およびその言語の仕様に従ったリンク方法でコレポDLL版印刷エンジンをリンクしてお使い頂けますようお願い申し上げます。

1.2 配布可能ファイル

作成したアプリケーションがコレポ印刷エンジンにアクセスするためには、以下のランタイムファイルを含める必要があります。以下のランタイムファイルはコレポ正規ユーザーに限り、作成したアプリケーションに含めて配布することが可能です。

配布可能なダイナミックリンクライブラリ

CWREPENG.DLL 及び CWBARCODE.DLL

また、コレポ印刷データファイルを参照する必要がある場合は、コレポ印刷データファイル参照用コレポ・ビューアー(ビューアー-本体もしくはインストーラ)を作成したアプリケーションに含めて配布することが可能です。

配布可能なビューアー・アプリケーション

コレポ・ビューアー CRPVIEW.exe

コレポ・ビューアー・インストーラ CRVINST.exe

(コレポインストールCDのCRDVIEWフォルダにCRVINST.EXEがあります)

1.3 プログラミングの概要

レポート印刷エンジンの制御は3つのセッションに分けられます。

印刷エンジン・セッション

印刷エンジン・セッションは、印刷エンジンのオープン (cwOpenEngine) で始まり、印刷エンジンのクローズ (cwCloseEngine) で終わります。

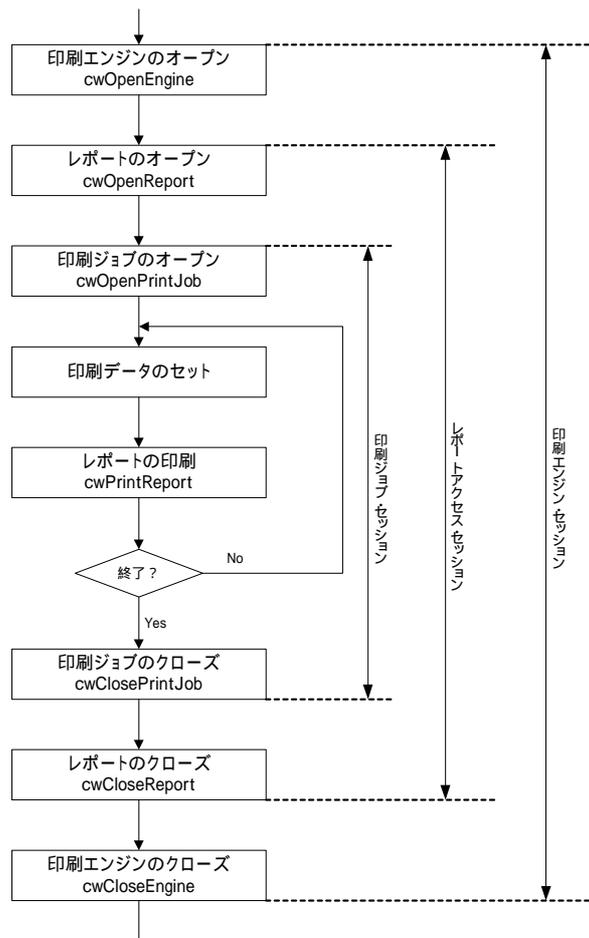
レポートアクセス・セッション

レポートアクセス・セッションはレポートのオープン (cwOpenReport) で始まり、レポートのクローズ (cwCloseReport) で終わります。

印刷ジョブ・セッション

印刷ジョブ・セッションは、印刷ジョブのオープン (cwOpenPrintJob) で始まり、印刷ジョブのクローズ (cwClosePrintJob) で終わります。

以下に示すのは、簡単な印刷エンジンのプログラミングの流れです。



1.4 印刷エンジン関数の一覧

コレポDLL版印刷エンジンの関数の一覧です。

印刷エンジン・セッションの関数

関数名	説明
cwOpenEngine	印刷エンジンを初期化します。
cwCloseEngine	印刷エンジンを終了します。
cwGetVersion	印刷エンジンのバージョンを取得します。
cwSetErrorMode	エラーモードを設定します。
cwGetLastError	エラーコードを取得します。
cwPrinterSetupDlg	プリンタの設定ダイアログを表示します。
cwSetInternetParam	インターネットに接続するためのユーザー情報を設定します。

レポートアクセス・セッションの関数

関数名	説明
基本的な関数	
cwOpenReport	レポートをロードします。
cwOpenPrintFile	印刷データファイルをロードします。
cwCloseReport	レポートを開放します。
cwSelectFmtPage	レポートの操作対象となるレイアウトページを指定します。
cwSetOffset	レポート内の全アイテムを指定オフセットぶん移動させます。
cwSetText	テキストまたはバーコード・アイテムに印刷する文字列を設定します。
cwGetText	テキストまたはバーコード・アイテムに設定されている文字列を取得します。
cwSetIntToStr	符号付き INT 型データを数字文字列に変換して、テキスト・アイテムに印刷する文字列として設定します。
cwSetUIntToStr	符号なし INT 型データを数字文字列に変換して、テキスト・アイテムに印刷する文字列として設定します。
cwSetLongToStr	符号付き LONG 型データを数字文字列に変換して、テキスト・アイテムに印刷する文字列として設定します。
cwSetULongToStr	符号なし LONG 型データを数字文字列に変換して、テキスト・アイテムに印刷する文字列として設定します。
cwSetDoubleToStr	Double 型データ(64Bit)を数字文字列に変換して、テキスト・アイテムに印刷する文字列として設定します。

cwSetTextByGrpIdx	テキストまたはバーコード・アイテムに印刷する文字列を設定します (インデックス付きグループ用)。
cwGetTextByGrpIdx	テキストまたはバーコードアイテムに設定されている文字列を取得します (インデックス付きグループ用)。
cwSetIntToStrByGrpIdx	符号付き INT 型データを数字文字列に変換して、テキスト・アイテムに印刷する文字列として設定します (インデックス付きグループ用)。
cwSetUIntToStrByGrpIdx	符号なし INT 型データを数字文字列に変換して、テキスト・アイテムに印刷する文字列として設定します (インデックス付きグループ用)。
cwSetLongToStrByGrpIdx	符号付き LONG 型データを数字文字列に変換して、テキスト・アイテムに印刷する文字列として設定します (インデックス付きグループ用)。
cwSetULongToStrByGrpIdx	符号なし LONG 型データを数字文字列に変換して、テキスト・アイテムに印刷する文字列として設定します (インデックス付きグループ用)。
cwSetDoubleToStrByGrpIdx	Double 型データ (64Bit) を数字文字列に変換して、テキスト・アイテムに印刷する文字列として設定します (インデックス付きグループ用)。

用紙・プリンタ名アクセス関数

cwSetPageDef	用紙の設定 (用紙サイズ・給紙方法)・長さの単位を設定します。
cwGetPageDef	用紙の設定 (用紙サイズ・給紙方法)・長さの単位を取得します。
cwSetPrinterName	プリンタ名を設定します。
cwGetPrinterName	プリンタ名を取得します。
cwSetDEVMODE	DEVMODE をプリンタに設定します。

アイテムプロパティ操作関数

cwSetImagePath	イメージ・アイテムにイメージをロードします。
cwSetImageDIB	イメージをイメージ・アイテムにコピーします。
cwSetDIBits	ビットマップをイメージ・アイテムに設定します。
cwSetDPI	イメージの解像度を設定します。
cwClearItemData	文字列/バーコード/イメージ・アイテムの印刷データを空にします。
cwClearAllLabel	全てのテキスト・アイテムの印刷用文字列を空にします。

<code>cwClearAllBarcode</code>	全てのバーコード・アイテムの印刷用バーコードイメージを空にします。
<code>cwClearAllImage</code>	全てのイメージ・アイテムの印刷用イメージを空にします。
<code>cwSetBound</code>	アイテムの位置を設定します。
<code>cwGetBound</code>	アイテムの位置を取得します。
<code>cwSetPen</code>	アイテムの線の太さ・線種（実線/点線等）・線色・角の丸みを設定します。
<code>cwGetPen</code>	アイテムの線の太さ・線種（実線/点線等）・線色・角の丸みの設定を取得します。
<code>cwSetBrush</code>	アイテムの背景の色・塗りつぶし模様を設定します。
<code>cwGetBrush</code>	アイテムの背景の色・塗りつぶし模様を取得します。
<code>cwSetFont</code>	アイテムのフォント情報を設定します。
<code>cwGetFont</code>	アイテムのフォント情報を取得します。
<code>cwSetMultiLine</code>	アイテムのマルチラインモードを取得します。
<code>cwGetMultiLine</code>	アイテムのマルチラインモードを設定します。
<code>cwSetTextAlign</code>	アイテムの文字の寄せを設定します。
<code>cwGetTextAlign</code>	アイテムの文字の寄せを取得します。
<code>cwSetSpace</code>	アイテムの文字間・行間を設定します。
<code>cwGetSpace</code>	アイテムの文字間・行間を取得します。
<code>cwSetTextEscapement</code>	アイテムの文字の横書き/縦書きを設定します。
<code>cwGetTextEscapement</code>	アイテムの文字の横書き/縦書きを取得します。
<code>cwGetTextMetrics</code>	テキストアイテムの印刷する文字列の高さ・幅を取得します。
<code>cwSetWordwrap</code>	アイテムのテキストの折り返しを設定します。
<code>cwGetWordwrap</code>	アイテムのテキストの折り返しの設定を取得します。
<code>cwSetBarcodeProp</code>	バーコード・アイテムのプロパティを設定します。
<code>cwGetBarcodeProp</code>	バーコードのプロパティを取得します。
<code>cwSetImgStretch</code>	イメージ・アイテムのストレッチモードを設定します。
<code>cwGetImgStretch</code>	イメージ・アイテムのストレッチモードを取得します。
<code>cwSetVisible</code>	アイテムの表示・非表示を設定します。
<code>cwGetVisible</code>	アイテムの表示・非表示を取得します。
<code>cwGrpSetVisible</code>	グループの表示・非表示を設定します。
<code>cwSetForeground</code>	アイテムの前面・背面への移動を設定します。
<code>cwGetItemType</code>	アイテムの種別を取得します。
<code>cwRenameItem</code>	アイテムのグループ名及び名前を変えます。
<code>cwSetFixed</code>	アイテムを固定項目とするか否かを設定します。
<code>cwGetFixed</code>	アイテムを固定項目とするか否かの設定を取得します。

アイテムリピートモード関数

<code>cwStartRepeat</code>	アイテムまたは、グループをリピートモードにします。
<code>cwDupBoxLine</code>	Box/Line アイテムの複製を作ります。
<code>cwDupGrp</code>	グループの複製を作ります。
<code>cwCheckDupBound</code>	複製されるグループまたはアイテムがレポートの範囲内に作成できるか否かを確認します。

アイテム一覧取得関数

<code>cwGetItemList</code>	アイテムの名前の一覧の取得します。
<code>cwGetFirstItem</code>	選択されているページ内の先頭アイテムのグループ名・アイテム名・種別を取得します。
<code>cwGetNextItem</code>	直前の <code>cwGetFirstItem</code> に続けて、次のアイテムののグループ名・アイテム名・種別を取得します。
<code>cwCloseItemHandle</code>	<code>cwGetFirstItem</code> で取得したハンドルを開放します <code>cwCloseItemHandle</code>

インデックス付きグループのアイテムプロパティ操作関数

<code>cwSetImagePathByGrpIdx</code>	イメージ・アイテムにイメージをロードします。
<code>cwSetImageDIBByGrpIdx</code>	イメージをイメージ・アイテムにコピーします。
<code>cwSetDIBitsByGrpIdx</code>	ビットマップをイメージ・アイテムに設定します。
<code>cwSetDPIByGrpIdx</code>	イメージの解像度を設定します。
<code>cwClearItemDataByGrpIdx</code>	文字列/バーコード/イメージ・アイテムの印刷データを空にします。
<code>cwSetBoundByGrpIdx</code>	アイテムの位置を設定します。
<code>cwGetBoundByGrpIdx</code>	アイテムの位置を取得します。
<code>cwSetPenByGrpIdx</code>	アイテムの線の太さ・線種(実線/点線等)・線色・角の丸みを設定します。
<code>cwGetPenByGrpIdx</code>	アイテムの線の太さ・線種(実線/点線等)・線色・角の丸みの設定を取得します。
<code>cwSetBrushByGrpIdx</code>	アイテムの背景の色・塗りつぶし模様を設定します。
<code>cwGetBrushByGrpIdx</code>	アイテムの背景の色・塗りつぶし模様を取得します。
<code>cwSetFontByGrpIdx</code>	アイテムの Font 情報を設定します。
<code>cwGetFontByGrpIdx</code>	アイテムの Font 情報を取得します。
<code>cwSetTextAlignByGrpIdx</code>	アイテムの文字の寄せを設定します。
<code>cwGetTextAlignByGrpIdx</code>	アイテムの文字の寄せを取得します。
<code>cwSetSpaceByGrpIdx</code>	アイテムの文字間・行間を設定します。
<code>cwGetSpaceByGrpIdx</code>	アイテムの文字間・行間を取得します。
<code>cwSetTextEscapementByGrpIdx</code>	アイテムの文字の横書き/縦書きを設定します。
<code>cwGetTextEscapementByGrpIdx</code>	アイテムの文字の横書き/縦書きを取得します。
<code>cwGetTextMetricsByGrpIdx</code>	テキストアイテムの印刷する文字列の高さ・幅を取得します。
<code>cwSetWordwrapByGrpIdx</code>	アイテムのテキストの折り返しを設定します。
<code>cwGetWordwrapByGrpIdx</code>	アイテムのテキストの折り返しの設定を取得します。
<code>cwSetBarcodePropByGrpIdx</code>	バーコード・アイテムのプロパティを設定します。
<code>cwGetBarcodePropByGrpIdx</code>	バーコード・アイテムのプロパティを取得します。
<code>cwSetImgStretchByGrpIdx</code>	イメージ・アイテムのストレッチモードを設定します。
<code>cwGetImgStretchByGrpIdx</code>	イメージ・アイテムのストレッチモードを取得します。
<code>cwSetVisibleByGrpIdx</code>	アイテムの表示・非表示を設定します。
<code>cwGetVisibleByGrpIdx</code>	アイテムの表示・非表示を取得します。
<code>cwSetForegroundByGrpIdx</code>	アイテムの前面・背面への移動を設定します。
<code>cwSetFixedByGrpIdx</code>	アイテムの固定項目を設定します。
<code>cwGetFixedByGrpIdx</code>	アイテムを固定項目の設定を取得します。

レポートの生成 / アイテムの生成・削除 関数

cwCreateReport	レポートデータを印刷エンジン内部に生成します。
cwCreateItem	アイテムを生成します。
cwDeleteItem	アイテムを削除します。

印刷ジョブ・セッションの関数

関数名	説明
cwOpenPrintJob	印刷ジョブを開始します。
cwOpenPrintJobEx	印刷部数指定して、印刷ジョブを開始します。
cwClosePrintJob	印刷ジョブを終了します。
cwShowPreView	プレビュー画面を表示します。
cwPrintReport	レポートを印刷します。
cwGetPrvhWnd	プレビューウィンドウのハンドルを取得します。
cwPrvhWndStatus	プレビューウィンドウが開いているか否かの状態を取得します。
cwSetPrvhWndBounds	プレビューウィンドウの位置/サイズを設定します。
cwGetPrvhWndBounds	プレビューウィンドウの位置/サイズを取得します。
cwClosePrvhWnd	プレビューウィンドウを閉じます。
cwSetPrvhWndPage	プレビューウィンドウに表示するページ番号を設定します。
cwGetPrvhWndPage	プレビューウィンドウに表示されているページ番号を取得します。
cwSetPrvhWndScale	プレビューウィンドウに表示する倍率を設定します。
cwGetPrvhWndScale	プレビューウィンドウに表示されている倍率を取得します。
cwSetPrvhWndBarStatus	プレビューウィンドウのツールバー・ステータスバーの表示/非表示を設定します。
cwGetPrvhWndBarStatus	プレビューウィンドウのツールバー・ステータスバーの表示/非表示の設定を取得します。
cwPrvhWndDoPrint	プレビューウィンドウの印刷ボタン押下と同様に印刷を実行します。

第 2 章

関数リファレンス

この章は、コレポDLL版印刷エンジンのI/Fを紹介します。

2.1 印刷エンジンセッションの関数

cwOpenEngine

レポート印刷エンジンに要求を受け付ける準備をさせます。

C/C++

```
BOOL cwOpenEngine(void);
```

VB

```
Declare Function cwOpenEngine Lib "cwRepEng.dll"()As Long
```

Delphi

```
function cwOpenEngine():Integer;stdcall;external 'cwRepEng.dll';
```

パラメータ

パラメータは有りません。

戻り値

関数が成功すると、0以外の値が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError関数を使います。

参照

cwCloseEngine

cwCloseEngine

レポート印刷エンジンを終了させます。

C/C++

```
BOOL cwCloseEngine(void);
```

VB

```
Declare Function cwCloseEngine Lib "cwRepEng.dll" () As Long
```

Delphi

```
function cwCloseEngine():Integer;stdcall;external  
'cwRepEng.dll';
```

パラメータ

パラメータは有りません。

戻り値

関数が成功すると、0以外の値が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

[cwOpenEngine](#)

cwGetVersion

レポート印刷エンジンのバージョンを取得します。

C/C++

```
DWORD cwGetVersion(void);
```

VB

```
Declare Function cwGetVersion Lib "cwRepEng.dll" () As Long
```

Delphi

```
function cwGetVersion(): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

パラメータは有りません。

戻り値

レポート印刷エンジンのバージョンを返します。

cwSetErrorMode

エラーが発生したときに、印刷エンジンにエラーダイアログボックス表示を任せるか、呼び出し側のアプリケーションが処理するかを設定します。デフォルトは、1で、エラーに関するメッセージボックスを表示せず、呼び出し側プロセスへそのエラーを送信します。

C/C++

```
UINT cwSetErrorMode(  
    UINT uMode    //プロセスのエラーモード  
);
```

VB

```
Declare Function cwSetErrorMode Lib "cwRepEng.dll" (ByVal uMode As  
Long) As Long
```

Delphi

```
function cwSetErrorMode(uMode: Integer): Integer; stdcall; external  
'cwRepEng.dll';
```

パラメータ

uMode

エラーモードを指定します。次の値を指定します。

値	動作
---	----

0	印刷エンジンがエラーダイアログボックスを表示します。
---	----------------------------

1	エラーに関するメッセージボックスを表示せず、呼び出し側プロセスへそのエラーを送信します。(デフォルト)
---	---

戻り値

以前のエラーモードが返ります。

cwGetLastError

印刷エンジンが持つ最新のエラーコードを取得します。

C/C++

```
DWORD cwGetLastError(  
    DWORD*WinErr  
);
```

VB

```
Declare Function cwGetLastError Lib "cwRepEng.dll" (WinErr As  
Long) As Long
```

Delphi

```
function cwGetLastError(WinErr: Integer): Integer; stdcall; external  
'cwRepEng.dll';
```

パラメータ

WinErr

Windows のエラーコードを保存するポインタを指定します。

戻り値が 268435456 (1000000) の場合、システムエラーであることを示します。この場合、WinErr には、Windows のエラーコードが返ります。

括弧内の値は 16 進数値です。

戻り値

印刷エンジンが持つ最新のエラーコードが返ります。

cwPrinterSetupDlg

プリンタ及び用紙の設定ダイアログを表示します。

C/C++

```
BOOL cwPrinterSetupDlg (  
    LPCSTR IpPrinerName,    //プリンタ名  
    CW_PAGETYPE *IpPageDef //ページ・プロパティ・データ  
);
```

VB

```
Declare Function cwPrinterSetupDlg Lib "cwRepEng.dll"(ByVal  
    IpPrinerName As String,hCwIitm As CW_PAGETYPE )As Long
```

Delphi

```
function cwPrinterSetupDlg(IpPrinerName:Pchar;hCwIitm:Pointer):  
    Integer;stdcall;external 'cwRepEng.dll';
```

パラメータ

IpPrinerName

プリンタ名を格納するバッファへのポインタを指定します。バッファの大きさは32バイトを必要とします。

デフォルトのプリンタを設定指定したい場合は、バッファの内容を NULL でクリアしてください。また、特定のプリンタを設定したい場合は、そのプリンタ名をバッファに設定して呼び出してください。

IpPageDef

ページ・プロパティ・データ(CW_PAGETYPE)のポインタを指定します。

戻り値

関数が成功すると、0以外の値が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetPageDef , cwGetPageDef , cwSetPrinterName , cwGetPrinterName

cwSetInternetParam

レポートファイル及び、イメージデータをインターネット上のサーバーから読み込む際のユーザー情報を設定します。

C/C++

```
BOOL cwSetInternetParam (
    LPCSTR IpProxyServer,    //プロキシサーバー名
    LPCSTR IpUserName,      //ユーザー名
    LPCSTR IpPassword,      //パスワード
    INT numPort              //ポート番号
);
```

VB

```
Declare Function cwSetInternetParam Lib "cwRepEng.dll" (ByVal
IpProxyServer As String, ByVal IpUserName As String, ByVal
IpPassword As String, ByVal numPort As Long )As Long
```

Delphi

```
function cwSetInternetParam(IpProxyServer:PChar; IpUserName:
PChar; IpPassword:PChar; numPort: Integer): Integer; stdcall; external
'cwRepEng.dll';
```

パラメータ

IpProxyServer

http 転送で特定のプロキシサーバを経由する場合、そのプロキシサーバ名が入った、NULL で終わる文字列へのポインタを指定します。必要のない場合は NULL または "" を指定します。

IpUserName

http 転送でプロキシサーバーがユーザー名を必要とする場合、または、ftp 転送でユーザー名を必要とする場合、ユーザー名が入った、NULL で終わる文字列へのポインタを指定します。必要のない場合は NULL または "" を指定します。

ftp 転送で、ユーザー名を指定しない場合、接続時のユーザー名に "anonymous" が使われます。

IpPassword

ユーザー名を指定した場合で、http 転送でプロキシサーバーがパスワードを必要とする場合、または、ftp 転送でパスワードを必要とする場合、パスワードが入った、NULL で終わる文字列へのポインタを指定します。必要のない場合は NULL または "" を指定します。

ftp 転送で、パスワードを指定しない場合、接続時のパスワードに E-Mail の既定アカウントのパスワードが使われます。

numPort

接続するサーバー側のポート番号を指定します。0 を指定するとデフォルトのポート番号が使われます。

戻り値

関数が成功すると、0以外の値が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwOpenReport` , `cwSetImagePath`

cwOpenReport

レポートデータを印刷エンジン内部に読み込み、対象レポートを特定するハンドルを返します。

C/C++

```
HCWREP cwOpenReport(  
    LPCSTR lpFileName    //レポートファイル名  
);
```

VB

```
Declare Function cwOpenReport Lib "cwRepEng.dll"(ByVal lpFileName  
As String)As Long
```

Delphi

```
function cwOpenReport(lpFileName:PChar):Integer;stdcall;external  
'cwRepEng.dll';
```

パラメータ

lpFileName

対象のレポートファイルの名前（パス）を保持している、NULL で終わる文字列へのポインタを指定します。

http 転送でレポートファイルを開く場合は、レポートファイルへの URL を保持している、NULL で終わる文字列へのポインタを指定します。

ftp 転送でレポートファイルを開く場合は、以下に示す形式でレポートファイルへのパスを保持している、NULL で終わる文字列へのポインタを指定します。

ftp 転送の記述形式

ftp://(ftpサーバー名、又は IP アドレス)/(レポートファイルへのパス)
と記述します。

例) ftp サーバー ftp.com-ware.jp で ./Sample/Test.crp を開く場合。

ftp://ftp.com-ware.jp/Sample/Test.crp
となります。

戻り値

関数が成功すると、対象レポートを特定するハンドルを返します。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwCloseReport , cwSetInternetParam

cwOpenPrintFile

印刷 JOB 関数で作成した印刷データファイルを印刷エンジン内部に読み込み、印刷準備を行います。対象レポートを特定するハンドルを返します。

C/C++

```
HCWREP cwOpenPrintFile(  
    LPCSTR lpFileName //印刷データファイル名  
);
```

VB

```
Declare Function cwOpenPrintFile Lib "cwRepEng.dll"(ByVal  
lpFileName As String)As Long
```

Delphi

```
function cwOpenPrintFile(lpFileName:PChar):Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

lpFileName

対象の印刷データファイルの名前(パス)を保持している、NULLで終わる文字列へのポインタを指定します。

http 転送で印刷データファイルを開く場合は、印刷データファイルへの URL を保持している、NULLで終わる文字列へのポインタを指定します。

ftp 転送で印刷データファイルを開く場合は、以下に示す形式で印刷データファイルへのパスを保持している、NULLで終わる文字列へのポインタを指定します。

ftp 転送の記述形式

ftp://(ftpサーバー名、又はIPアドレス)/(レポートファイルへのパス)
と記述します。

例) ftpサーバーftp.com-ware.jpで./Sample/Test.crdを開く場合。

ftp://ftp.com-ware.jp/Sample/Test.crd
となります。

戻り値

関数が成功すると、対象レポートを特定するハンドルを返します。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError関数を使います。

参照

cwCloseReport , cwSetInternetParam, cwOpenPrintJob

cwCloseReport

印刷エンジン内部に読み込んだレポートデータを破棄します。

C/C++

```
BOOL cwCloseReport(  
    HCREP hCwRep    //レポートハンドル  
);
```

VB

```
Declare Function cwCloseReport Lib "cwRepEng.dll" (ByVal hCwRep As  
Long) As Long
```

Delphi

```
function cwCloseReport(hCwRep: Integer): Integer; stdcall; external  
'cwRepEng.dll';
```

パラメータ

hCwRep

破棄したいレポートのハンドルを指定します。

戻り値

関数が成功すると、0以外の値を返します。関数が失敗すると、0が返ります。
エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwOpenReport

cwSelectFmtPage

アイテム制御関数で操作の対象となるレポート内のレイアウトページを選択します。

C/C++

```
BOOL cwSelectFmtPage(  
    HCWREP hCwRep ,      //レポートのハンドル  
    INT numPage          //レポートレイアウトページ番号  
);
```

VB

```
Declare Function cwSelectFmtPage Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal numPage As Long ) As Long
```

Delphi

```
function cwSelectFmtPage(hCwRep: Integer; numPage: Integer):  
Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

対象のレポートのハンドルを指定します。

numPage

対象レポートのレイアウトページ番号 (0 から 9) を指定します。1 ページ目が 0、2 ページ目が 1 となります。

戻り値

関数が成功すると、0 以外の値を返します。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

cwSetOffset

レポート内の全てのアイテムを指定オフセットぶん移動させます。使用するプリンタの特性による印刷ズレの微調整などに使います。

C/C++

```
BOOL cwSetOffset(  
    HCREP    hCwRep,        //レポートハンドル  
    DOUBLE   dbLeftOffset, //左を基準にした水平方向のオフセット  
    DOUBLE   dbTopOffset,  //上を基準にした垂直方向のオフセット  
);
```

VB

```
Declare Function cwSetOffset Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal dbLeftOffset As Double, ByVal dbTopOffset As Double) As  
Long
```

Delphi

```
function cwSetOffset(hCwRep: Integer; dbLeftOffset: Double;  
dbTopOffset: Double): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

dbLeftOffset

左を基準にした水平方向のオフセットをセットします。単位はレポートの長さの単位に従い、ミリメートル又はインチとなります。

dbTopOffset

上を基準にした垂直方向のオフセットをセットします。単位はレポートの長さの単位に従い、ミリメートル又はインチとなります。

戻り値

関数が成功すると、0以外の値が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

2.2 レポートアクセス・セッションの関数

2.2.1 基本的な関数

cwSetText

指定のテキスト・アイテム / バーコード・アイテムに指定の文字列を設定します。

C/C++

```
INT cwSetText(  
    HCWREP hCwRep,    //レポートのハンドル  
    LPCSTR lpGrpName, //グループ名  
    LPCSTR lpItemName, //アイテム名  
    LPCSTR lpText     //文字列  
);
```

VB

```
Declare Function cwSetText Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String, ByVal  
lpText As String) As Long
```

Delphi

```
function cwSetText(hCwRep: Integer; lpGrpName: PChar; lpItemName:  
PChar; lpText: PChar): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定
します。

lpItemName

アイテム名へのポインタを指定します。

lpText

設定する文字列へのポインタを指定します。(終端は NULL になっている必要が
あります)

戻り値

関数が成功すると、文字列の長さが返ります。関数が失敗すると、-1 が返ります。
エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetTextByGrpIdx , cwGetText , cwGetTextByGrpIdx , cwSetIntToStr ,

cwSetIntToStrByGrpIdx , cwSetUIntToStr ,cwSetUIntToStrByGrpIdx ,
cwSetDoubleToStr , cwSetDoubleToStrByGrpIdx, cwSetBarCodeProp ,
cwSetBarCodePropByGrpIdx , cwGetBarCodeProp ,
cwGetBarCodePropByGrpIdx

cwGetText

指定のテキスト・アイテム/バーコード・アイテムに設定されている文字列を取得します。

C/C++

```
INT cwGetText(  
    HCWREP hCwRep,        //レポートのハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,   //アイテム名  
    LPCSTR lpText        //文字列  
);
```

VB

```
Declare Function cwGetText Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String, ByVal  
lpText As String) As Long
```

Delphi

```
function cwGetText(hCwRep: Integer; lpGrpName: PChar; lpItemName:  
PChar; lpText: Pchar ): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

lpText

取得する文字列を格納するバッファへのポインタを指定します。NULL を指定すると文字列の長さ（バイト長）が返されます。

戻り値

関数が成功すると、文字列の長さが返ります。関数が失敗すると、-1 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetText , cwSetTextByGrpIdx , cwGetTextByGrpIdx , cwSetIntToStr ,
cwSetIntToStrByGrpIdx , cwSetUIntToStr , cwSetUIntToStrByGrpIdx ,
cwSetDoubleToStr , cwSetDoubleToStrByGrpIdx , cwSetBarCodeProp ,
cwSetBarCodePropByGrpIdx , cwGetBarCodeProp ,
cwGetBarCodePropByGrpIdx

cwSetIntToStr

INT 型の数値を指定の形式に変換して、指定のテキスト・アイテムに設定します。

C/C++

```
INT cwSetIntToStr(  
    HCWREP hCwRep,           //レポートのハンドル  
    LPCSTR IpGrpName,       //グループ名  
    LPCSTR IpItemName,      //アイテム名  
    LPCSTR IpHeadStr,       //前に付加する文字列  
    LPCSTR IpTailStr,       //後ろ付加する文字列  
    INT nVal,               //設定する数  
    UINT nFlags              //編集方法の指定  
);
```

VB

```
Declare Function cwSetIntToStr Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal IpGrpName As String, ByVal IpItemName As String, ByVal  
IpHeadStr As String, ByVal IpTailStr As String, ByVal nVal As  
Long, ByVal nFlags As Long) As Long
```

Delphi

```
function cwSetIntToStr(hCwRep: Integer; IpGrpName: PChar; IpItemName:  
PChar; IpHeadStr: PChar; IpTailStr: PChar; nVal: Integer; nFlags: Integer): Integer;  
stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

IpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定
します。

IpItemName

アイテム名へのポインタを指定します。

IpHeadStr

変換した数値文字列の前に文字列を付けたい場合、その付け加える文字列を指
定します。付け加えない場合は NULL を指定します。

IpTailStr

変換した数値文字列の後ろに文字列を付けたい場合、その付け加える文字列を
指定します。付け加えない場合は NULL を指定します。

nVal

設定する数値。

nFlags

数字の変換方法 以下のいずれかを指定します。

CW_EDIT_HAN (0) 半角数字文字列に変換します。

CW_EDIT_ZEN (1) 全角数字文字列に変換します

CW_EDIT_KNJ1(2) 漢数字多角文字に変換します。(壹, 弐, 参, 四..., 拾...)

CW_EDIT_KNJ2(3) 漢数字多角文字に変換します。(壹, 弐, 参, 四..., 十...)

CW_EDIT_KNJ3(4) 漢数字に変換します。(一, 二, 三, 四..., 十...)

CW_EDIT_KNJ4(5) 漢数字に変換します。千・万などの位をあらわす漢数字は付きません。(〇, 一, 二, 三, 四..., 九)

上記の値の1つと、以下の値の組合せで、数値の編集形式を指定します。

CW_EDIT_COMMA(16) カンマ編集します。(漢数字の場合は無効となります)

CW_EDIT_ZERO2SPC(32) 設定する値が0の場合、0の代わりに空白文字をセットします。

CW_EDIT_SIGNTAIL(64) 符号を後ろにつけます。

CW_EDIT_SIGNJPN(128) マイナス符号を に変換します。

戻り値

関数が成功すると、文字列の長さが返ります。関数が失敗すると、-1 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetText , cwSetTextByGrpIdx , cwGetText , cwGetTextByGrpIdx
cwSetIntToStrByGrpIdx , cwSetUIntToStr , cwSetUIntToStrByGrpIdx
cwSetDoubleToStr , cwSetDoubleToStrByGrpIdx

cwSetUIntToStr

UINT 型の数値を指定の形式に変換して、指定のテキスト・アイテムに設定します。

C/C++

```
INT cwSetUIntToStr(  
    HCWREP hCwRep,           //レポートのハンドル  
    LPCSTR IpGrpName,       //グループ名  
    LPCSTR IpItemName,     //アイテム名  
    LPCSTR IpHeadStr,      //前に付加する文字列  
    LPCSTR IpTailStr,      //後ろ付加する文字列  
    INT nVal,              //設定する数  
    UINT nFlags            //編集方法の指定  
);
```

VB

```
Declare Function cwSetUIntToStr Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal IpGrpName As String, ByVal IpItemName As String, ByVal  
IpHeadStr As String, ByVal IpTailStr As String, ByVal nVal As  
Long, ByVal nFlags As Long )As Long
```

Delphi

```
function cwSetUIntToStr(hCwRep: Integer; IpGrpName: PChar; IpItemName:  
PChar; IpHeadStr: PChar; IpTailStr: PChar; nVal: Integer; nFlags: Integer): Integer;  
stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

IpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定
します。

IpItemName

アイテム名へのポインタを指定します。

IpHeadStr

変換した数値文字列の前に文字列を付けたい場合、その付け加える文字列を指
定します。付け加えない場合は NULL を指定します。

IpTailStr

変換した数値文字列の後ろに文字列を付けたい場合、その付け加える文字列を
指定します。付け加えない場合は NULL を指定します。

nVal

設定する数値。

nFlags

数字の変換方法 以下のいずれかを指定します。

CW_EDIT_HAN (0)	半角数字文字列に変換します。
CW_EDIT_ZEN (1)	全角数字文字列に変換します
CW_EDIT_KNJ1(2)	漢数字多角文字に変換します。(壹, 弍, 参, 四..., 拾...)
CW_EDIT_KNJ2(3)	漢数字多角文字に変換します。(壹, 弍, 参, 四..., 十...)
CW_EDIT_KNJ3(4)	漢数字に変換します。(一, 二, 三, 四..., 十...)
CW_EDIT_KNJ4(5)	漢数字に変換します。千・万などの位をあらわす漢数字は付きません。(〇, 一, 二, 三, 四..., 九)

上記の値の1つと、以下の値の組合せで、数値の編集形式を指定します。

CW_EDIT_COMMA(16)	カンマ編集します。(漢数字の場合は無効となります)
CW_EDIT_ZERO2SPC(32)	設定する値が0の場合、0の代わりに空白文字をセットします。
CW_EDIT_SIGNTAIL(64)	符号を後ろにつけます。
CW_EDIT_SIGNJPN(128)	マイナス符号を に変換します。

戻り値

関数が成功すると、文字列の長さが返ります。関数が失敗すると、-1 が返ります。エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetText` , `cwSetTextByGrpIdx` , `cwGetText` , `cwGetTextByGrpIdx`
`cwSetIntToStr` , `cwSetIntToStrByGrpIdx` , `cwSetUIntToStrByGrpIdx`
`cwSetDoubleToStr` , `cwSetDoubleToStrByGrpIdx`

cwSetLongToStr

LONG 型の数値を指定の形式に変換して、指定のテキスト・アイテムに設定します。

C/C++

```
INT cwSetLongToStr(  
    HCWREP hCwRep,           //レポートのハンドル  
    LPCSTR IpGrpName,       //グループ名  
    LPCSTR IpItemName,     //アイテム名  
    LPCSTR IpHeadStr,      //前に付加する文字列  
    LPCSTR IpTailStr,      //後ろ付加する文字列  
    INT nVal,              //設定する数  
    UINT nFlags            //編集方法の指定  
);
```

VB

```
Declare Function cwSetLongToStr Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal IpGrpName As String, ByVal IpItemName As String, ByVal  
IpHeadStr As String, ByVal IpTailStr As String, ByVal nVal As  
Long, ByVal nFlags As Long) As Long
```

Delphi

```
function cwSetLongToStr(hCwRep: Integer; IpGrpName: PChar; IpItemName:  
PChar; IpHeadStr: PChar; IpTailStr: PChar; nVal: Integer; nFlags: Integer): Integer;  
stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

IpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定
します。

IpItemName

アイテム名へのポインタを指定します。

IpHeadStr

変換した数値文字列の前に文字列を付けたい場合、その付け加える文字列を指
定します。付け加えない場合は NULL を指定します。

IpTailStr

変換した数値文字列の後ろに文字列を付けたい場合、その付け加える文字列を
指定します。付け加えない場合は NULL を指定します。

nVal

設定する数値。

nFlags

数字の変換方法 以下のいずれかを指定します。

CW_EDIT_HAN (0) 半角数字文字列に変換します。

CW_EDIT_ZEN (1) 全角数字文字列に変換します

CW_EDIT_KNJ1(2) 漢数字多角文字に変換します。(壹, 弍, 参, 四..., 拾...)

CW_EDIT_KNJ2(3) 漢数字多角文字に変換します。(壹, 弍, 参, 四..., 十...)

CW_EDIT_KNJ3(4) 漢数字に変換します。(一, 二, 三, 四..., 十...)

CW_EDIT_KNJ4(5) 漢数字に変換します。千・万などの位を問わず漢数字は付きません。(〇, 一, 二, 三, 四..., 九)

上記の値の1つと、以下の値の組合せで、数値の編集形式を指定します。

CW_EDIT_COMMA(16) カンマ編集します。(漢数字の場合は無効となります)

CW_EDIT_ZERO2SPC(32) 設定する値が0の場合、0の代わりに空白文字をセットします。

CW_EDIT_SIGNTAIL(64) 符号を後ろにつけます。

CW_EDIT_SIGNJPN(128) マイナス符号を に変換します。

戻り値

関数が成功すると、文字列の長さが返ります。関数が失敗すると、-1 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetLongToStrByGrpIdx , cwSetULongToStr , cwSetULongToStrByGrpIdx

cwSetULongToStr

ULONG 型の数値を指定の形式に変換して、指定のテキスト・アイテムに設定します。

C/C++

```
INT cwSetULongToStr(  
    HCWREP hCwRep,           //レポートのハンドル  
    LPCSTR IpGrpName,       //グループ名  
    LPCSTR IpItemName,      //アイテム名  
    LPCSTR IpHeadStr,       //前に付加する文字列  
    LPCSTR IpTailStr,       //後ろ付加する文字列  
    INT nVal,               //設定する数  
    UINT nFlags              //編集方法の指定  
);
```

VB

```
Declare Function cwSetULongToStr Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal IpGrpName As String, ByVal IpItemName As String, ByVal  
IpHeadStr As String, ByVal IpTailStr As String, ByVal nVal As  
Long, ByVal nFlags As Long )As Long
```

Delphi

```
function cwSetULongToStr(hCwRep: Integer; IpGrpName: PChar; IpItemName:  
PChar; IpHeadStr: PChar; IpTailStr: PChar; nVal: Integer; nFlags: Integer): Integer;  
stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

IpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定
します。

IpItemName

アイテム名へのポインタを指定します。

IpHeadStr

変換した数値文字列の前に文字列を付けたい場合、その付け加える文字列を指
定します。付け加えない場合は NULL を指定します。

IpTailStr

変換した数値文字列の後ろに文字列を付けたい場合、その付け加える文字列を
指定します。付け加えない場合は NULL を指定します。

nVal

設定する数値。

nFlags

数字の変換方法 以下のいずれかを指定します。

CW_EDIT_HAN (0) 半角数字文字列に変換します。

CW_EDIT_ZEN (1) 全角数字文字列に変換します

CW_EDIT_KNJ1(2) 漢数字多角文字に変換します。(壹, 弍, 参, 四..., 拾...)

CW_EDIT_KNJ2(3) 漢数字多角文字に変換します。(壹, 弍, 参, 四..., 十...)

CW_EDIT_KNJ3(4) 漢数字に変換します。(一, 二, 三, 四..., 十...)

CW_EDIT_KNJ4(5) 漢数字に変換します。千・万などの位をあらわす漢数字は付きません。(〇, 一, 二, 三, 四..., 九)

上記の値の1つと、以下の値の組合せで、数値の編集形式を指定します。

CW_EDIT_COMMA(16) カンマ編集します。(漢数字の場合は無効となります)

CW_EDIT_ZERO2SPC(32) 設定する値が0の場合、0の代わりに空白文字をセットします。

CW_EDIT_SIGNTAIL(64) 符号を後ろにつけます。

CW_EDIT_SIGNJPN(128) マイナス符号を に変換します。

戻り値

関数が成功すると、文字列の長さが返ります。関数が失敗すると、-1 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetLongToStr , cwSetLongToStrByGrpIdx , cwSetULongToStrByGrpIdx

cwSetDoubleToStr

DOUBLE 型の数値を指定の形式に変換して、指定のテキスト・アイテムに設定します。

C/C++

```
INT cwSetDoubleToStr(  
    HCWREP hCwRep,        //レポートのハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,    //アイテム名  
    LPCSTR lpHeadStr,     //前に付加する文字列  
    LPCSTR lpTailStr,     //後ろ付加する文字列  
    DOUBLE nVal,          //設定する数値  
    INT nDecimal,         //少数点以下の桁数  
    UINT nFlags            //編集方法の指定  
);
```

VB

```
Declare Function cwSetDoubleToStr Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal lpGrpName As String, ByVal lpItemName As String, ByVal  
lpHeadStr As String, ByVal lpTailStr As String, ByVal dbVal As  
Double, ByVal nDecimal As Long, ByVal nFlags As Long) As Long
```

Delphi

```
function cwSetDoubleToStr(hCwRep: Integer; lpGrpName: PChar; lpItemName:  
PChar; lpHeadStr: PChar; lpTailStr: PChar; dbVal: Double; nDecimal: Integer;  
nFlags: Integer): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

lpHeadStr

変換した数値文字列の前に文字列を付けたい場合、その付け加える文字列を指定します。付け加えない場合は NULL を指定します。

lpTailStr

変換した数値文字列の後ろに文字列を付けたい場合、その付け加える文字列を指定します。付け加えない場合は NULL を指定します。

nVal

設定する数値。

nDecimal

少数点以下の桁数を指定します。

nFlags

有効桁数の丸め方法、以下のいずれかを指定します。

CW_EDIT_DBL_ROUNDOFF 四捨五入で有効桁数を丸めます。
(256)

CW_EDIT_DBL_CUT(512) 切り捨てで有効桁数を丸めます。

CW_EDIT_DBL_RAISE 切り上げで有効桁数を丸めます。
(1024)

数字の変換方法 以下のいずれかを指定します。

CW_EDIT_HAN (0) 半角数字文字列に変換します。

CW_EDIT_ZEN (1) 全角数字文字列に変換します

CW_EDIT_KNJ1(2) 漢数字多角文字に変換します。(壹, 弐, 参, 四..., 拾...)

CW_EDIT_KNJ2(3) 漢数字多角文字に変換します。(壹, 弐, 参, 四..., 十...)

CW_EDIT_KNJ3(4) 漢数字に変換します。(一, 二, 三, 四..., 十...)

CW_EDIT_KNJ4(5) 漢数字に変換します。千・万などの位をあらわす漢数字は付きません。(〇, 一, 二, 三, 四..., 九)

上記の値の1つと、以下の値の組合せで、数値の編集形式を指定します。

CW_EDIT_COMMA(16) カンマ編集します。(漢数字の場合は無効となります)

CW_EDIT_ZERO2SPC(32) 設定する値が0の場合、0の代わりに空白文字をセットします。

CW_EDIT_SIGNTAIL(64) 符号を後ろにつけます。

CW_EDIT_SIGNJPN(128) マイナス符号を に変換します。

戻り値

関数が成功すると、文字列の長さが返ります。関数が失敗すると、-1 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetText , cwSetTextByGrpIdx , cwGetText , cwGetTextByGrpIdx

cwSetIntToStr , cwSetIntToStrByGrpIdx , cwSetUIntToStr

cwSetUIntToStrByGrpIdx , cwSetDoubleToStrByGrpIdx

cwSetTextByGrpIdx

指定のテキスト・アイテム/バーコード・アイテムに指定の文字列を設定します。この関数は、デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
INT cwSetTextByGrpIdx(  
    HCWREP hCwRep,           //レポートのハンドル  
    LPCSTR lpGrpName,       //グループ名  
    LPCSTR lpItemName,      //アイテム名  
    LPCSTR lpText,          //文字列  
    INT nDigit,             //インデックスの桁数  
    INT nIdx                //インデックス  
);
```

VB

```
Declare Function cwSetTextByGrpIdx Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal lpGrpName As String, ByVal lpItemName As String, ByVal  
lpText As String, ByVal nDigit As Long, ByVal nIdx As Long) As Long
```

Delphi

```
function cwSetTextByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; lpText: PChar; nDigit: Integer; nIdx: Integer ):  
Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

lpText

設定する文字列へのポインタを指定します。(終端は NULL になっている必要があります)

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、文字列の長さが返ります。関数が失敗すると、-1 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetText , cwGetText , cwGetTextByGrpIdx , cwSetIntToStr
cwSetIntToStrByGrpIdx , cwSetUIntToStr , cwSetUIntToStrByGrpIdx ,
cwSetDoubleToStr , cwSetDoubleToStrByGrpIdx

cwGetTextByGrpIdx

指定のテキスト・アイテム/バーコード・アイテムに設定されている文字列を取得します。この関数は、デザイナーで、グループの複製をした時に付けたインデックス(添え字)部分を変数として扱えるようになっています。

C/C++

```
INT cwGetTextByGrpIdx(  
    HCWREP hCwRep,      //レポートのハンドル  
    LPCSTR lpGrpName,   //グループ名  
    LPCSTR lpItemName,  //アイテム名  
    LPCSTR lpText,      //文字列  
    INT nDigit,         //インデックスの桁数  
    INT nIndex         //インデックス  
);
```

VB

```
Declare Function cwGetTextByGrpIdx Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal lpGrpName As String, ByVal lpItemName As String, ByVal  
lpText As String, ByVal nDigit As Long, ByVal nIndex As Long )As Long
```

Delphi

```
function cwGetTextByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; lpText: PChar; nDigit: Integer; nIndex: Integer ):  
Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

lpText

取得する文字列を格納するバッファへのポインタを指定します。NULL を指定すると文字列の長さ(バイト長)が返されます。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、文字列の長さが返ります。関数が失敗すると、-1 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetText, cwSetTextByGrpIdx , cwGetText, cwSetIntToStr,
cwSetIntToStrByGrpIdx ,cwSetUIntToStr , cwSetUIntToStrByGrpIdx
cwSetDoubleToStr, cwSetDoubleToStrByGrpIdx

cwSetIntToStrByGrpIdx

INT 型の数値を指定の形式に変換して、指定のテキスト・アイテムに設定します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
INT cwSetIntToStrByGrpIdx(  
    HCWREP hCwRep,          //レポートのハンドル  
    LPCSTR lpGrpName,      //グループ名  
    LPCSTR lpItemName,     //アイテム名  
    LPCSTR lpHeadStr,     //前に付加する文字列  
    LPCSTR lpTailStr,     //後ろ付加する文字列  
    INT nVal,              //設定する数  
    UINT nFlags,          //編集方法の指定  
    INT nDigit,           //インデックスの桁数  
    INT nIdx              //インデックス  
);
```

VB

```
Declare Function cwSetIntToStrByGrpIdx Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, ByVal lpHeadStr As String, ByVal lpTailStr As String, ByVal  
    nVal As Long, ByVal nFlags As Long, ByVal nDigit As Long, ByVal nIdx  
    As Long) As Long
```

Delphi

```
function cwSetIntToStrByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; lpHeadStr: PChar; lpTailStr: PChar; nVal: Integer;  
    nFlags: Integer; nDigit: Integer; nIdx: Integer ): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

lpHeadStr

変換した数値文字列の前に文字列を付けたい場合、その付け加える文字列を指定します。付け加えない場合は NULL を指定します。

lpTailStr

変換した数値文字列の後ろに文字列を付けたい場合、その付け加える文字列を指定します。付け加えない場合は NULL を指定します。

nVal

設定する数値。

nFlags

数字の変換方法 以下のいずれかを指定します。

CW_EDIT_HAN (0) 半角数字文字列に変換します。

CW_EDIT_ZEN (1) 全角数字文字列に変換します

CW_EDIT_KNJ1(2) 漢数字多角文字に変換します。(壹, 弍, 参, 四..., 拾...)

CW_EDIT_KNJ2(3) 漢数字多角文字に変換します。(壹, 弍, 参, 四..., 十...)

CW_EDIT_KNJ3(4) 漢数字に変換します。(一, 二, 三, 四..., 十...)

CW_EDIT_KNJ4(5) 漢数字に変換します。千・万などの位を問わず漢数字は付きません。(〇, 一, 二, 三, 四..., 九)

上記の値の1つと、以下の値の組合せで、数値の編集形式を指定します。

CW_EDIT_COMMA(16) カンマ編集します。(漢数字の場合は無効となります)

CW_EDIT_ZERO2SPC(32) 設定する値が0の場合、0の代わりに空白文字をセットします。

CW_EDIT_SIGNTAIL(64) 符号を後ろにつけます。

CW_EDIT_SIGNJPN(128) マイナス符号を に変換します。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、文字列の長さが返ります。関数が失敗すると、-1が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetText , cwSetTextByGrpIdx , cwGetText , cwGetTextByGrpIdx

cwSetIntToStr , cwSetUIntToStr , cwSetUIntToStrByGrpIdx

cwSetDoubleToStr , cwSetDoubleToStrByGrpIdx

cwSetUIntToStrByGrpIdx

UINT 型の数値を指定の形式に変換して、指定のテキスト・アイテムに設定します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
INT cwSetUIntToStrByGrpIdx(  
    HCWREP hCwRep,          //レポートのハンドル  
    LPCSTR lpGrpName,      //グループ名  
    LPCSTR lpItemName,     //アイテム名  
    LPCSTR lpHeadStr,     //前に付加する文字列  
    LPCSTR lpTailStr,     //後ろ付加する文字列  
    UINT nVal,             //設定する数  
    UINT nFlags,          //編集方法の指定  
    INT nDigit,           //インデックスの桁数  
    INT nIdx              //インデックス  
);
```

VB

```
Declare Function cwSetUIntToStrByGrpIdx Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, ByVal lpHeadStr As String, ByVal lpTailStr As String, ByVal  
    nVal As Long, ByVal nFlags As Long, ByVal nDigit As Long, ByVal nIdx  
    As Long ) As Long
```

Delphi

```
function cwSetUIntToStrByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; lpHeadStr: PChar; lpTailStr: PChar; nVal: Cardinal;  
    nFlags: Integer; nDigit: Integer; nIdx: Integer ): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

lpHeadStr

変換した数値文字列の前に文字列を付けたい場合、その付け加える文字列を指定します。付け加えない場合は NULL を指定します。

lpTailStr

変換した数値文字列の後ろに文字列を付けたい場合、その付け加える文字列を指定します。付け加えない場合は NULL を指定します。

nVal

設定する数値。

nFlags

数字の変換方法 以下のいずれかを指定します。

CW_EDIT_HAN (0) 半角数字文字列に変換します。

CW_EDIT_ZEN (1) 全角数字文字列に変換します

CW_EDIT_KNJ1(2) 漢数字多角文字に変換します。(壹, 弐, 参, 四..., 拾...)

CW_EDIT_KNJ2(3) 漢数字多角文字に変換します。(壹, 弐, 参, 四..., 十...)

CW_EDIT_KNJ3(4) 漢数字に変換します。(一, 二, 三, 四..., 十...)

CW_EDIT_KNJ4(5) 漢数字に変換します。千・万などの位を問わず漢数字は付きません。(〇, 一, 二, 三, 四..., 九)

上記の値の1つと、以下の値の組合せで、数値の編集形式を指定します。

CW_EDIT_COMMA(16) カンマ編集します。(漢数字の場合は無効となります)

CW_EDIT_ZERO2SPC(32) 設定する値が0の場合、0の代わりに空白文字をセットします。

CW_EDIT_SIGNTAIL(64) 符号を後ろにつけます。

CW_EDIT_SIGNJPN(128) マイナス符号を に変換します。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、文字列の長さが返ります。関数が失敗すると、-1が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetText , cwSetTextByGrpIdx , cwGetText , cwGetTextByGrpIdx

cwSetIntToStr , cwSetIntToStrByGrpIdx , cwSetUIntToStr

cwSetDoubleToStr , cwSetDoubleToStrByGrpIdx

cwSetLongToStrByGrpIdx

LONG 型の数値を指定の形式に変換して、指定のテキスト・アイテムに設定します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
INT cwSetLongToStrByGrpIdx(  
    HCWREP hCwRep,          //レポートのハンドル  
    LPCSTR lpGrpName,      //グループ名  
    LPCSTR lpItemName,     //アイテム名  
    LPCSTR lpHeadStr,     //前に付加する文字列  
    LPCSTR lpTailStr,     //後ろ付加する文字列  
    LONG nVal,            //設定する数  
    UINT nFlags,         //編集方法の指定  
    INT nDigit,          //インデックスの桁数  
    INT nIndex           //インデックス  
);
```

VB

```
Declare Function cwSetLongToStrByGrpIdx Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, ByVal lpHeadStr As String, ByVal lpTailStr As String, ByVal  
    nVal As Long, ByVal nFlags As Long, ByVal nDigit As Long, ByVal nIndex  
    As Long) As Long
```

Delphi

```
function cwSetLongToStrByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; lpHeadStr: PChar; lpTailStr: PChar; nVal: LongInt;  
    nFlags: Integer; nDigit: Integer; nIndex: Integer): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

lpHeadStr

変換した数値文字列の前に文字列を付けたい場合、その付け加える文字列を指定します。付け加えない場合は NULL を指定します。

lpTailStr

変換した数値文字列の後ろに文字列を付けたい場合、その付け加える文字列を指定します。付け加えない場合は NULL を指定します。

nVal

設定する数値。

nFlags

数字の変換方法 以下のいずれかを指定します。

CW_EDIT_HAN (0) 半角数字文字列に変換します。

CW_EDIT_ZEN (1) 全角数字文字列に変換します

CW_EDIT_KNJ1(2) 漢数字多角文字に変換します。(壹, 弍, 参, 四..., 拾...)

CW_EDIT_KNJ2(3) 漢数字多角文字に変換します。(壹, 弍, 参, 四..., 十...)

CW_EDIT_KNJ3(4) 漢数字に変換します。(一, 二, 三, 四..., 十...)

CW_EDIT_KNJ4(5) 漢数字に変換します。千・万などの位を問わず漢数字は付きません。(〇, 一, 二, 三, 四..., 九)

上記の値の 1 つと、以下の値の組合せで、数値の編集形式を指定します。

CW_EDIT_COMMA(16) カンマ編集します。(漢数字の場合は無効となります)

CW_EDIT_ZERO2SPC(32) 設定する値が 0 の場合、0 の代わりに空白文字をセットします。

CW_EDIT_SIGNTAIL(64) 符号を後ろにつけます。

CW_EDIT_SIGNJPN(128) マイナス符号を に変換します。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0 を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、文字列の長さが返ります。関数が失敗すると、-1 が返ります。

エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetLongToStr , cwSetULongToStr , cwSetULongToStrByGrpIdx

cwSetULongToStrByGrpIdx

ULONG 型の数値を指定の形式に変換して、指定のテキスト・アイテムに設定します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
INT cwSetULongToStrByGrpIdx(  
    HCWREP hCwRep,          //レポートのハンドル  
    LPCSTR lpGrpName,      //グループ名  
    LPCSTR lpItemName,     //アイテム名  
    LPCSTR lpHeadStr,     //前に付加する文字列  
    LPCSTR lpTailStr,     //後ろ付加する文字列  
    ULONG nVal,           //設定する数  
    UINT nFlags,         //編集方法の指定  
    INT nDigit,          //インデックスの桁数  
    INT nIdx             //インデックス  
);
```

VB

```
Declare Function cwSetULongToStrByGrpIdx Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, ByVal lpHeadStr As String, ByVal lpTailStr As String, ByVal  
    nVal As Long, ByVal nFlags As Long, ByVal nDigit As Long, ByVal nIdx  
    As Long ) As Long
```

Delphi

```
function cwSetULongToStrByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; lpHeadStr: PChar; lpTailStr: PChar; nVal: Cardinal;  
    nFlags: Integer; nDigit: Integer; nIdx: Integer ): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

lpHeadStr

変換した数値文字列の前に文字列を付けたい場合、その付け加える文字列を指定します。付け加えない場合は NULL を指定します。

lpTailStr

変換した数値文字列の後ろに文字列を付けたい場合、その付け加える文字列を指定します。付け加えない場合は NULL を指定します。

nVal

設定する数値。

nFlags

数字の変換方法 以下のいずれかを指定します。

CW_EDIT_HAN (0) 半角数字文字列に変換します。

CW_EDIT_ZEN (1) 全角数字文字列に変換します

CW_EDIT_KNJ1(2) 漢数字多角文字に変換します。(壹, 弐, 参, 四..., 拾...)

CW_EDIT_KNJ2(3) 漢数字多角文字に変換します。(壹, 弐, 参, 四..., 十...)

CW_EDIT_KNJ3(4) 漢数字に変換します。(一, 二, 三, 四..., 十...)

CW_EDIT_KNJ4(5) 漢数字に変換します。千・万などの位を問わず漢数字は付きません。(〇, 一, 二, 三, 四..., 九)

上記の値の1つと、以下の値の組合せで、数値の編集形式を指定します。

CW_EDIT_COMMA(16) カンマ編集します。(漢数字の場合は無効となります)

CW_EDIT_ZERO2SPC(32) 設定する値が0の場合、0の代わりに空白文字をセットします。

CW_EDIT_SIGNTAIL(64) 符号を後ろにつけます。

CW_EDIT_SIGNJPN(128) マイナス符号を に変換します。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、文字列の長さが返ります。関数が失敗すると、-1が返ります。エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetLongToStr` , `cwSetLongToStrByGrpIdx` , `cwSetULongToStr`

cwSetDoubleToStrByGrpIdx

DOUBLE 型の数値を指定の形式に変換して、指定のテキスト・アイテムに設定します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
INT cwSetDoubleToStrByGrpIdx(  
    HCWREP hCwRep,          //レポートのハンドル  
    LPCSTR lpGrpName,       //グループ名  
    LPCSTR lpItemName,     //アイテム名  
    LPCSTR lpHeadStr,      //前に付加する文字列  
    LPCSTR lpTailStr,      //後ろ付加する文字列  
    DOUBLE nVal,           //設定する数値  
    INT nDecimal,          //少数点以下の桁数  
    UINT nFlags,           //編集方法の指定  
    INT nDigit,            //インデックスの桁数  
    INT nIdx                //インデックス  
);
```

VB

```
Declare Function cwSetDoubleToStrByGrpIdx Lib "cwRepEng.dll"  
(ByVal hCwRep As Long,ByVal lpGrpName As String,ByVal lpItemName  
As String,ByVal lpHeadStr As String,ByVal lpTailStr As String,ByVal  
dbVal As Double,ByVal nDecimal As Long,ByVal nFlags As Long,ByVal  
nDigit As Long,ByVal nIdx As Long)As Long
```

Delphi

```
function cwSetDoubleToStrByGrpIdx(hCwRep: Integer; lpGrpName:  
PChar; lpItemName: PChar; lpHeadStr: PChar; lpTailStr: PChar;  
dbVal: Double; nDecimal: Integer; nFlags: Integer; nDigit: Integer;  
nIdx: Integer): Integer;stdcall;external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

lpHeadStr

変換した数値文字列の前に文字列を付けたい場合、その付け加える文字列を指定します。付け加えない場合は NULL を指定します。

lpTailStr

変換した数値文字列の後ろに文字列を付けたい場合、その付け加える文字列を

指定します。付け加えない場合は NULL を指定します。

nVal

設定する数値。

nDecimal

少数点以下の桁数を指定します。

nFlags

有効桁数の丸め方法、以下のいずれかを指定します。

CW_EDIT_DBL_ROUND OFF 四捨五入で有効桁数を丸めます。
(256)

CW_EDIT_DBL_CUT(512) 切り捨てで有効桁数を丸めます。

CW_EDIT_DBL_RAISE 切り上げで有効桁数を丸めます。
(1024)

数字の変換方法 以下のいずれかを指定します。

CW_EDIT_HAN (0) 半角数字文字列に変換します。

CW_EDIT_ZEN (1) 全角数字文字列に変換します

CW_EDIT_KNJ1(2) 漢数字多角文字に変換します。(壹, 弍, 参, 四..., 拾...)

CW_EDIT_KNJ2(3) 漢数字多角文字に変換します。(壹, 弍, 参, 四..., 十...)

CW_EDIT_KNJ3(4) 漢数字に変換します。(一, 二, 三, 四..., 十...)

CW_EDIT_KNJ4(5) 漢数字に変換します。千・万などの位を問わず漢数字は付きません。(〇, 一, 二, 三, 四..., 九)

上記の値の1つと、以下の値の組合せで、数値の編集形式を指定します。

CW_EDIT_COMMA(16) カンマ編集します。(漢数字の場合は無効となります)

CW_EDIT_ZERO2SPC(32) 設定する値が0の場合、0の代わりに空白文字をセットします。

CW_EDIT_SIGNTAIL(64) 符号を後ろにつけます。

CW_EDIT_SIGNJPN(128) マイナス符号を に変換します。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、文字列の長さが返ります。関数が失敗すると、-1が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetText , cwSetTextByGrpIdx , cwGetText , cwGetTextByGrpIdx ,
cwSetIntToStr , cwSetIntToStrByGrpIdx , cwSetUIntToStr ,
cwSetUIntToStrByGrpIdx , cwSetDoubleToStr

2.2.2 用紙・プリンタ名アクセス関数

cwSetPageDef

レポートの用紙サイズ・給紙方法を設定します。

C/C++

```
BOOL cwSetPageDef (
    HCREP hCwRep           //レポートハンドル
    CW_PAGDEF *lpPageDef   //ページ・プロパティ・データ
);
```

VB

```
Declare Function cwSetPageDef Lib "cwRepEng.dll" (ByVal hCwRep As Long, lpPageDef As CW_PAGDEF) As Long
```

Delphi

```
function cwSetPageDef(hCwRep: Integer; lpPageDef: Pointer): Integer;
stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpPageDef

ページ・プロパティ・データ (CW_PAGDEF) のポインタを指定します。通常、cwPrinterSetupDlg 関数で取得したページ定義データを指定します。

戻り値

関数が成功すると、0以外の値が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwPrinterSetupDlg , cwGetPageDef , cwSetPrinterName ,
cwGetPrinterName, cwSetDEVMODE

cwGetPageDef

レポートの用紙サイズ・給紙方法の設定を取得します。

C/C++

```
BOOL cwGetPageDef (  
    HCREP hCwRep           //レポートハンドル  
    CW_PAGDEF *lpPageDef  //ページ・プロパティ・データ  
);
```

VB

```
Declare Function cwGetPageDef Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, lpPageDef As CW_PAGDEF )As Long
```

Delphi

```
function cwGetPageDef(hCwRep: Integer; lpPageDef: Pointer): Integer;  
stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpPageDef

ページ・プロパティ・データ(CW_PAGDEF)を格納するバッファのポインタを指定します。

戻り値

関数が成功すると、0以外の値が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError関数を使います。

参照

cwPrinterSetupDlg , cwSetPageDef , cwSetPrinterName ,
cwGetPrinterName, cwSetDEVMODE

cwSetPrinterName

レポートを印刷するプリンタの名前を設定します。

C/C++

```
INT cwSetPrinterName(  
    HCREP hCwRep,           //レポートハンドル  
    LPCSTR lpPrinterName   //プリンタ名  
);
```

VB

```
Declare Function cwSetPrinterName Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal lpPrinterName As String) As Long
```

Delphi

```
function cwSetPrinterName(hCwRep: Integer; lpPrinterName: PChar ):  
Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpPrinterName

設定するプリンタ名の入った文字列のポインタ。(文字列の終端は NULL になっている必要があります)

戻り値

関数が成功すると、プリンタ名の文字列の長さを返します。関数が失敗すると、-1 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwPrinterSetupDlg , cwSetPageDef , cwGetPageDef , cwGetPrinterName,
cwSetDEVMODE

cwGetPrinterName

レポートを印刷するプリンタの名前を取得します。

C/C++

```
INT cwGetPrinterName(  
    HCWREP hCwRep,          //レポートハンドル  
    LPCSTR lpPrinterName //プリンタ名  
);
```

VB

```
Declare Function cwGetPrinterName Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal lpPrinterName As String) As Long
```

Delphi

```
function cwGetPrinterName(hCwRep: Integer; lpPrinterName: PChar ):  
Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpPrinterName

取得するプリンタ名を格納するバッファのポインタ (文字列の終端は NULL になっています)。バッファサイズは最大で 32 文字分必要となります。

戻り値

関数が成功すると、プリンタ名の文字列の長さを返します。関数が失敗すると、-1 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwPrinterSetupDlg , cwSetPageDef , cwGetPageDef , cwSetPrinterName,
cwSetDEVMODE

cwSetDEVMODE

レポートを印刷するプリンタの名前を取得します。

C/C++

```
INT cwSetDEVMODE(  
    HCWREP hCwRep,    //レポートハンドル  
    LPDEVMODE lpDevMode    //DEVMODE 構造体  
);
```

VB

```
Declare Function cwSetDEVMODE Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpDevMode As Long) As Long
```

Delphi

```
function cwSetDEVMODE(hCwRep: Integer; lpDevMode: Pointer):  
Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpDevMode

DEVMODE 構造体へのポインタを指定します (DEVMODE の詳細に関しては、Windows GDI を参照してください)。

戻り値

関数が成功すると、プリンタ名の文字列の長さを返します。関数が失敗すると、-1 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwPrinterSetupDlg , cwSetPageDef , cwGetPageDef , cwSetPrinterName,
cwGetPrinterName

2.2.3 アイテムプロパティ操作関数

cwSetImagePath

イメージ・アイテムに指定のイメージファイルからイメージを読み込みます。

C/C++

```
BOOL cwSetImagePath(  
    HCWREP hCwRep,          //レポートのハンドル  
    LPCSTR lpGrpName,      //グループ名  
    LPCSTR lpItemName,     //アイテム名  
    LPCSTR lpFileName      //イメージファイルのパス  
);
```

VB

```
Declare Function cwSetImagePath Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String, ByVal  
lpFileName As String) As Long
```

Delphi

```
function cwSetImagePath(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; lpFileName: PChar ): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定
します。

lpItemName

アイテム名へのポインタを指定します。

lpFileName

イメージファイルへのパスが格納されている文字列のポインタ。(イメージ
ファイルは、TIFF/JPEG/BMP)

http 転送でイメージファイルを開く場合は、イメージファイルへの URL を保
持している、NULL で終わる文字列へのポインタを指定します。

ftp 転送でイメージファイルを開く場合は、以下に示す形式でイメージファ
イルへのパスを保持している、NULL で終わる文字列へのポインタを指定します。

ftp 転送の記述形式

ftp://(ftpサーバー名、又は IP アドレス)/(イメージファイルへのパス)

と記述します。

例) ftp サーバ-ftp.com-ware.jp で./Sample/Test.jpeg を開く場合。
ftp://ftp.com-ware.jp/Sample/Test.jpeg
となります。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetImagePathByGrpIdx , cwSetImageDIB , cwSetImageDIBByBrpIdx ,
cwClearItemData , cwClearItemDataByGrpIdx ,cwSetInternetParam ,
cwSetImgStretch , cwSetImgStretchByGrpIdx , cwGetImgStretch
cwGetImgStretchByGrpIdx,cwSetDIBits,cwSetDIBitsByGrpIdx,cwSetDPI,
cwSetDPIByGrpIdx

cwSet ImageDIB

イメージをイメージ・アイテムにコピーします。

C/C++

```
BOOL cwSet ImageDIB(  
    HCWREP hCwRep,           //レポートハンドル  
    LPCSTR lpGrpName,       //グループ名  
    LPCSTR lpItemName,     //アイテム名  
    HDC hDc,                //コピー元のデバイスコンテキストのハンドル  
    INT nXOriginSrc,       //コピー元の X 座標  
    INT nYOriginSrc,       //コピー元の Y 座標  
    INT nWidthSrc,         //コピー元の幅  
    INT nHeightSrc        //コピー元の高さ  
);
```

VB

```
Declare Function cwSet ImageDIB Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String, ByVal hDc  
As Long, ByVal nXOriginSrc As Long, ByVal nYOriginSrc As Long, ByVal  
nWidthSrc As Long, ByVal nHeightSrc As Long) As Long
```

Delphi

```
function cwSet ImageDIB(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; hDc: HDC; nXOriginSrc: Integer; nYOriginSrc: Integer;  
nWidthSrc: Integer; nHeightSrc: Integer): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

hDc

コピー元のデバイスコンテキストのハンドル。

XOriginSrc

コピー元の X 座標。

YOriginSrc

コピー元の Y 座標。

nWidth

コピー元の幅。

nHeight

コピー元の高さ。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetImagePath` , `cwSetImagePathByGrpIdx` , `cwSetImageDIBByBrpIdx` ,
`cwClearItemData` , `cwClearItemDataByGrpIdx` , `cwSetImgStrech` ,
`cwSetImgStrechByGrpIdx` , `cwGetImgStretch` , `cwGetImgStretchByGrpIdx` ,
`cwSetDIBits` , `cwSetDIBitsByGrpIdx` , `cwSetDPI` , `cwSetDPIByGrpIdx`

cwSetDIBits

ビットマップをイメージ・アイテムに設定します。

C/C++

```
BOOL cwSetDIBits(  
    HCWREP hCwRep,           //レポートハンドル  
    LPCSTR IpGrpName,       //グループ名  
    LPCSTR IpItemName,     //アイテム名  
    LPBITMAPINFO IpBinfo,  //ビットマップ情報  
    LPVOID IpBits           //ビットマップデータ  
);
```

VB

```
Public Declare Function cwSetDIBits Lib "cwRepEng.dll" (ByVal  
hCwRep As Long, ByVal IpGrpName As String, ByVal IpItemName As  
String, ByVal IpBinfo As Long, ByVal IpBitsData As Long ) As Long
```

Delphi

```
function cwSetDIBits(hCwRep: Integer; IpGrpName: PChar; IpItemName:  
PChar; IpBinfo: Pointer; IpBits: Pointer ): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

IpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

IpItemName

アイテム名へのポインタを指定します。

IpBinfo

DIB の情報を保持している BITMAPINFO 構造体へのポインタを指定します。

IpBits

DIB の色データ (ビットマップのビット配列) を保持しているバイト配列へのポインタを指定します。ビットマップ値の形式は、IpBinfo パラメータが指す BITMAPINFO 構造体の biBitCount メンバに依存します。なお、NULL (0) が指定された場合、パック済み DIB が指定されたものとみなして、処理が行われます。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetImagePath , cwSetImagePathByGrpIdx , cwSetImageDIBByBrpIdx ,
cwClearItemData , cwClearItemDataByGrpIdx , cwSetImgStrech ,
cwSetImgStrechByGrpIdx , cwGetImgStretch , cwGetImgStretchByGrpIdx
cwSetDIBitsByGrpIdx , cwSetDPI , cwSetDPIByGrpIdx

cwSetDPI

イメージ・アイテムの解像度 (DPI) を設定します。

C/C++

```
BOOL cwSetDPI(  
    HCWREP hCwRep, //レポートハンドル  
    LPCSTR lpGrpName, //グループ名  
    LPCSTR lpItemName, //アイテム名  
    INT nDpi //解像度  
);
```

VB

```
Public Declare Function cwSetDPI Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal lpGrpName As String, ByVal lpItemName As String, ByVal  
nDpi As Long) As Long
```

Delphi

```
function function cwSetDPI(hCwRep:Integer; lpGrpName:PChar;  
lpItemName:PChar; nDpi:Integer ):Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

nDpi

イメージの解像度を指定します。イメージアイテムのストレッチが False の場合に有効で、印刷時に解像度に従って適正な大きさにイメージを補正して出力します。0 を指定した場合は、出力するプリンタの解像度で印刷します。

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetImagePath , cwSetImagePathByGrpIdx , cwSetImageDIBByBrpIdx ,
cwClearItemData , cwClearItemDataByGrpIdx , cwSetImgStretch ,
cwSetImgStretchByGrpIdx , cwGetImgStretch , cwGetImgStretchByGrpIdx
cwSetDIBits, cwSetDIBitsByGrpIdx, cwSetDPIByGrpIdx

cwClearItemData

アイテムの印刷データを空にします。

テキスト・アイテム：文字列を空にします。

バーコード・アイテム：バーコード及びバーコードデータを空にします

イメージ・アイテム：イメージを空にします。

但し、Fixed(固定項目)が True に設定されているアイテムのデータはクリアされません。

C/C++

```
BOOL cwClearItemData(  
    HCWREP hCwRep,    //レポートのハンドル  
    LPCSTR lpGrpName, //グループ名  
    LPCSTR lpItemName //アイテム名  
);
```

VB

```
Declare Function cwClearItemData Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal lpGrpName As String, ByVal lpItemName As String) As  
Long
```

Delphi

```
function cwClearItemData(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar ): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwClearItemDataByGrpIdx , cwClearAllText , cwClearAllBarcode ,
cwClearAllImage

cwClearAllText

ページ内の全てのテキスト・アイテムに設定されている文字列を空にします。但し、Fixed(固定項目)がTrueに設定されているテキスト・アイテムの文字列は空にされません。

C/C++

```
BOOL cwClearAllText(  
    HCWREP hCwRep    //レポートハンドル  
);
```

VB

```
Declare Function cwClearAllText Lib "cwRepEng.dll" (ByVal hCwRep As  
Long) As Long
```

Delphi

```
function cwClearAllText(hCwRep: Integer): Integer;stdcall;external  
'cwRepEng.dll';
```

パラメータ

hCwRep
レポートのハンドルを指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError関数を使います。

参照

cwClearItemData , cwClearAllBarcode , cwClearAllImage

cwClearAllBarcode

ページ内の全てのバーコード・アイテムに設定されているバーコードデータを空にします。

但し、Fixed(固定項目)が True に設定されているバーコード・アイテムのバーコードデータは空にされません。

C/C++

```
BOOL cwClearAllBarcode(  
    HCWREP hCwRep    //レポートハンドル  
);
```

VB

```
Declare Function cwClearAllBarcode Lib "cwRepEng.dll" (ByVal hCwRep  
As Long )As Long
```

Delphi

```
function cwClearAllBarcode(hCwRep: Integer): Integer;stdcall  
;external 'cwRepEng.dll';
```

パラメータ

hCwRep
レポートのハンドルを指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwClearItemData , cwClearAllText , cwClearAllImage

cwClearAllImage

ページ内の全てのイメージ・アイテムに設定されているイメージを空にします。但し、Fixed(固定項目)がTrueに設定されているイメージ・アイテムのイメージは空にされません。

C/C++

```
BOOL cwClearAllImage(  
    HCWREP hCwRep    //レポートハンドル  
);
```

VB

```
Declare Function cwClearAllImageLib "cwRepEng.dll" (ByVal hCwRep As  
Long) As Long
```

Delphi

```
function cwClearAllImage(hCwRep: Integer): Integer;stdcall  
;external 'cwRepEng.dll';
```

パラメータ

hCwRep
レポートのハンドルを指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError関数を使います。

参照

cwClearItemData , cwClearAllText , cwClearAllBarcode

cwSetBound

アイテムの位置・大きさを設定します。

C/C++

```
BOOL cwSetBound(  
HCWREP hCwRep,          //レポートハンドル  
LPCSTR lpGrpName,      //グループ名  
LPCSTR lpItemName,     //アイテム名  
INT *nLeft,            //左位置  
INT *nTop,             //上位置  
INT *nWidth,           //幅  
INT *nHeight           //高さ  
);
```

VB

```
Declare Function cwSetBound Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String, nLeft As  
Long, nTop As Long, nWidth As Long, nHeight As Long) As Long
```

Delphi

```
function cwSetBound(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; nLeft: Pointer; nTop: Pointer; nWidth: Pointer;  
nHeight: Pointer): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

nLeft

左位置の値を格納している INT 型変数へのポインタ。設定しない場合は、NULL を指定します。

nTop

上位置の値を格納している INT 型変数へのポインタ。設定しない場合は、NULL を指定します。

nWidth

幅の値を格納している INT 型変数へのポインタ。設定しない場合は、NULL を指定します。

nHeight

高さの値を格納している INT 型変数へのポインタ。設定しない場合は、NULL を指定します。

位置・大きさの値は、レポートの長さの単位により異なります。
単位がミリメートルの場合、1/100 ミリメートルを基本単位とします。インチの
場合は、1 / 1 0 0 0 インチを基本単位とします。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetBoundByGrpIdx` , `cwGetBound` , `cwGetBoundByGrpIdx` , `cwSetVisible` ,
`cwSetVisibleByGrpIdx` , `cwGetVisible` , `cwGetVisibleByGrpIdx` ,
`cwSetForeground` , `cwSetForegroundByGrpIdx`

cwGetBound

アイテムの位置・大きさの設定を取得します。

C/C++

```
BOOL cwGetBound(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,   //アイテム名  
    INT *nLeft,          //左位置  
    INT *nTop,           //上位置  
    INT *nWidth,         //幅  
    INT *nHeight        //高さ  
);
```

VB

```
Declare Function cwGetBound Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String, nLeft As  
Long, nTop As Long, nWidth As Long, nHeight As Long) As Long
```

Delphi

```
function cwGetBound(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; nLeft: Pointer; nTop: Pointer; nWidth: Pointer;  
nHeight: Pointer): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

nLeft

左位置の値を格納する INT 型変数へのポインタ。取得しない場合は、NULL を指定します。

nTop

上位置の値を格納する INT 型変数へのポインタ。取得しない場合は、NULL を指定します。

nWidth

幅の値を格納する INT 型変数へのポインタ。取得しない場合は、NULL を指定します。

nHeight

高さの値を格納する INT 型変数へのポインタ。取得しない場合は、NULL を指定します。

位置・大きさの値は、レポートの長さの単位により異なります。
単位がミリメートルの場合、1/100 ミリメートルを基本単位とします。インチの場合、1 / 1000 インチを基本単位とします。

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。
エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetBound` , `cwSetBoundByGrpIdx` , `cwGetBoundByGrpIdx` , `cwSetVisible`
`cwSetVisibleByGrpIdx` , `cwGetVisible` , `cwGetVisibleByGrpIdx` ,
`cwSetForeground` , `cwSetForegroundByGrpIdx`

cwSetPen

アイテムの線や枠の設定をします。

C/C++

```
BOOL cwSetPen(  
    HCWREP hCwRep,           //レポートハンドル  
    LPCSTR lpGrpName,       //グループ名  
    LPCSTR lpItemName,     //アイテム名  
    INT *nPenStyle,        //線の模様  
    COLORREF *crPenColor,  //線の色  
    INT *nWidth,          //線の太さ  
    INT *nRound,          //枠の角の丸み  
    INT *nCapStyle        //線の終端の形  
);
```

VB

```
Declare Function cwSetPen Lib "cwRepEng.dll" (ByVal hCwRep As Long,  
ByVal lpGrpName As String,ByVal lpItemName As String,nPenStyle As  
Long,crPenColor As Long,nWidth As Long,nRound As Long,nCapStyle As  
Long )As Long
```

Delphi

```
function cwSetPen(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; nPenStyle: Pointer; crPenColor: Pointer;  
nWidth: Pointer; nRound: Pointer; nCapStyle: Pointer): Integer;stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

nPenStyle

線の模様を格納している INT 型へのポインタを指定します。設定しない場合は、NULL を指定します。以下の値のいずれかを指定します。

CW_LS_SOLID	(0)	実線
CW_LS_DASH	(1)	破線
CW_LS_DOT	(2)	点線
CW_LS_DASHDOT	(3)	一点鎖線
CW_LS_DASHDOTDOT	(4)	二点鎖線

crPenColor

線の色を格納している COLORREF 型へのポインタを指定します。設定しない場合は、NULL を指定します。

nWidth

線の太さの値を格納している INT 型変数へのポインタを指定します。単位は 1/100 ポイント単位です。設定しない場合は、NULL を指定します。

nRound

枠の角の丸みの値を格納している INT 型変数へのポインタを指定します。単位はレポートの長さの単位により異なります。単位がミリメートルの場合、1/100 ミリメートルを基本単位とします。インチの場合は、1 / 1000 インチを基本単位とします。設定しない場合は、NULL を指定します。

nCapStyle (Windows2000/XP のみ有効)

線の終端の形の値を格納している INT 型変数へのポインタを指定します。設定しない場合は、NULL を指定します。

CW_ENDCAP_ROUND	(0)	終端を丸くします。
CW_ENDCAP_SQUARE	(1)	終端を四角にします。
CW_ENDCAP_FLAT	(2)	終端を平らにします。

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetPenByGrpIdx , cwGetPen , cwGetPenByGrpIdx , cwSetBrush ,
cwSetBrushByGrpIdx , cwGetBrush , cwGetBrushByGrpIdx

cwGetPen

アイテムの線や枠の設定を取得します。

C/C++

```
BOOL cwGetPen(  
HCWREP hCwRep,          //レポートハンドル  
LPCSTR IpGrpName,      //グループ名  
LPCSTR IpItemName,     //アイテム名  
INT *nPenStyle,        //線の模様  
COLORREF *crPenColor,  //線の色  
INT *nWidth,           //線の太さ  
INT *nRound,           //枠の角の丸み  
INT *nCapStyle         //線の終端の形  
);
```

VB

```
Declare Function cwGetPen Lib "cwRepEng.dll" (ByVal hCwRep As Long,  
ByVal IpGrpName As String, ByVal IpItemName As String, nPenStyle As  
Long, crPenColor As Long, nWidth As Long, nRound As Long, nCapStyle As  
Long) As Long
```

Delphi

```
function cwGetPen(hCwRep: Integer; IpGrpName: PChar;  
IpItemName: PChar; nPenStyle: Pointer; crPenColor: Pointer;  
nWidth: Pointer; nRound: Pointer; nCapStyle: Pointer): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

IpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULLを指定します。

IpItemName

アイテム名へのポインタを指定します。

nPenStyle

線の模様を格納する INT 型へのポインタを指定します。取得しない場合は、NULLを指定します。以下のいずれかの値が取得できます。

CW_LS_SOLID	(0)	実線
CW_LS_DASH	(1)	破線
CW_LS_DOT	(2)	点線
CW_LS_DASHDOT	(3)	一点鎖線
CW_LS_DASHDOTDOT	(4)	二点鎖線

crPenColor

線の色を格納する COLORREF 型へのポインタを指定します。取得しない場合は、NULL を指定します。

nWidth

線の太さの値を格納する INT 型変数へのポインタを指定します。単位は 1/100 ポイント単位です。設取得しない場合は、NULL を指定します。

nRound

枠の角の丸みの値を格納する INT 型変数へのポインタを指定します。単位はレポートの長さの単位により異なります。単位がミリメートルの場合、1/100 ミリメートルを基本単位とします。インチの場合は、1 / 1000 インチを基本単位とします。取得しない場合は、NULL を指定します。

nCapStyle (Windows2000/XP のみ有効)

線の終端の形の値を格納する INT 型変数へのポインタを指定します。取得しない場合は、NULL を指定します。以下のいずれかの値が取得できます。

CW_ENDCAP_ROUND	(0)	終端を丸くします。
CW_ENDCAP_SQUARE	(1)	終端を四角にします。
CW_ENDCAP_FLAT	(2)	終端を平らにします。

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetPen , cwSetPenByGrpIdx , cwGetPenByGrpIdx , cwSetBrush ,
cwSetBrushByGrpIdx , cwGetBrush , cwGetBrushByGrpIdx

cwSetBrush

アイテムの背景の塗りつぶし模様や色の設定をします。

C/C++

```
BOOL cwSetBrush(  
    HCWREP    hCwRep,        //レポートハンドル  
    LPCSTR    lpGrpName,     //グループ名  
    LPCSTR    lpItemName,    //アイテム名  
    INT       *nBrushStyle,  //塗りつぶしの模様  
    COLORREF  *crBrushColor  //塗りつぶしの色  
);
```

VB

```
Declare Function cwSetBrush Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String,  
nBrushStyle As Long, crBrushColor As Long) As Long
```

Delphi

```
function cwSetBrush(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; nBrushStyle: Pointer; crBrushColor: Pointer ):  
Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

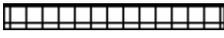
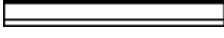
グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

nBrushStyle

塗りつぶしの模様を格納している INT 型へのポインタを指定します。設定しない場合は、NULL を指定します。以下の値のいずれかを指定します。

CW_BS_NONE	(0)	塗りつぶさない。
CW_BS_SOLID	(1)	塗りつぶす。
CW_BS_PATTERN1	(2)	
CW_BS_PATTERN2	(3)	
CW_BS_PATTERN3	(4)	
CW_BS_PATTERN4	(5)	
CW_BS_PATTERN5	(6)	
CW_BS_PATTERN6	(7)	
CW_BS_PATTERN7	(8)	
CW_BS_PATTERN8	(9)	

CW_BS_PATTERN9 (10)



CW_BS_PATTERN10 (11)



crBrushColor

塗りつぶしの色を格納している COLORREF 型へのポインタを指定します。設定しない場合は、NULL を指定します。

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetPen , cwSetPenByGrpIdx , cwGetPen , cwGetPenByGrpIdx ,
cwSetBrushByGrpIdx , cwGetBrush , cwGetBrushByGrpIdx

cwGetBrush

アイテムの背景の塗りつぶし模様や色の設定をします。

C/C++

```
BOOL cwGetBrush(  
    HCWREP    hCwRep,        //レポートハンドル  
    LPCSTR    lpGrpName,     //グループ名  
    LPCSTR    lpItemName,   //アイテム名  
    INT       *nBrushStyle, //塗りつぶしの模様  
    COLORREF  *crBrushColor //塗りつぶしの色  
);
```

VB

```
Declare Function cwGetBrush Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String,  
nBrushStyle As Long, crBrushColor As Long) As Long
```

Delphi

```
function cwGetBrush(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; nBrushStyle: Pointer; crBrushColor: Pointer ):  
Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

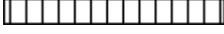
グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

nBrushStyle

塗りつぶしの模様を格納する INT 型へのポインタを指定します。取得しない場合は、NULL を指定します。以下の値のいずれかが取得できます。

CW_BS_SOLID	(1)	塗りつぶす。
CW_BS_PATTERN1	(2)	
CW_BS_PATTERN2	(3)	
CW_BS_PATTERN3	(4)	
CW_BS_PATTERN4	(5)	
CW_BS_PATTERN5	(6)	
CW_BS_PATTERN6	(7)	
CW_BS_PATTERN7	(8)	
CW_BS_PATTERN8	(9)	
CW_BS_PATTERN9	(10)	

CW_BS_PATTERN10 (11)



crBrushColor

塗りつぶしの色を格納する COLORREF 型へのポインタを指定します。取得しない場合は、NULL を指定します。

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetPen , cwSetPenByGrpIdx , cwGetPen , cwGetPenByGrpIdx ,
cwSetBrush , cwSetBrushByGrpIdx , cwGetBrushByGrpIdx

cwSetFont

テキスト・アイテムのフォント情報を設定をします。

C/C++

```
BOOL cwSetFont(  
HCWREP hCwRep, //レポートハンドル  
LPCSTR lpGrpName, //グループ名  
LPCSTR lpItemName, //アイテム名  
LPCSTR lpFontName, //フォント名  
INT *nFontStyle, //フォントスタイル  
INT *nFontSize, //フォントサイズ  
COLORREF *crFontColor //フォントの色  
);
```

VB

```
Declare Function cwSetFont Lib "cwRepEng.dll" (ByVal hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As String, ByVal lpFontName As String, nFontStyle As Long, nFontSize As Long, crFontColor As Long) As Long
```

Delphi

```
function cwSetFont(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; lpFontName: PChar; nFontStyle: Pointer;  
nFontSize: Pointer; crFontColor: Pointer ): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

lpFontName

フォント名へのポインタを指定します。設定しない場合は、NULL を指定します。

nFontStyle

フォントスタイルの値を格納している INT 型へのポインタを指定します。設定しない場合は、NULL を指定します。以下の値の組み合わせで設定します。

CW_FS_BOLD	(1)	太字
CW_FS_ITALIC	(2)	斜体
CW_FS_UNDERLINE	(4)	下線

nFontSize

フォントサイズを格納している INT 型へのポインタを指定します。設定しない

場合は、NULL を指定します。フォントサイズの単位は 1/10 ポイント単位で指定します。

`crFontColor`

フォント色を格納している COLORREF 型へのポインタを指定します。設定しない場合は、NULL を指定します。

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetFontByGrpIdx` , `cwGetFont` , `cwGetFontByGrpIdx` , `cwSetTextAlign` ,
`cwSetTextAlignByGrpIdx` , `cwGetTextAlign` , `cwGetTextAlignByGrpIdx` ,
`cwSetSpace` , `cwSetSpaceByGrpIdx` , `cwGetSpace` , `cwGetSpaceByGrpIdx` ,
`cwSetTextEscapement` , `cwSetTextEscapementByGrpIdx` , `cwGetTextEscapement` ,
`cwGetTextEscapementByGrpIdx` , `cwGetTextMetrics` , `cwGetTextMetricsByGrpIdx` ,
`cwSetWordwrap` , `cwGetWordwrap` , `cwSetWordwrapByGrpIdx` , `cwGetWordwrapByGrpIdx`

cwGetFont

テキスト・アイテムのフォント情報を取得します。

C/C++

```
BOOL cwGetFont(  
HCWREP hCwRep, //レポートハンドル  
LPCSTR lpGrpName, //グループ名  
LPCSTR lpItemName, //アイテム名  
LPCSTR lpFontName, //フォント名  
INT *nFontStyle, //フォントスタイル  
INT *nFontSize, //フォントサイズ  
COLORREF *crFontColor //フォントの色  
);
```

VB

```
Declare Function cwGetFont Lib "cwRepEng.dll" (ByVal hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As String, ByVal lpFontName As String, nFontStyle As Long, nFontSize As Long, crFontColor As Long) As Long
```

Delphi

```
function cwGetFont(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; lpFontName: PChar; nFontStyle: Pointer;  
nFontSize: Pointer; crFontColor: Pointer ): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

lpFontName

フォント名を格納するバッファのポインタを指定します。必要なバッファの大きさは65バイトです。取得しない場合は、NULL を指定します。

nFontStyle

フォントスタイルの値を格納する INT 型へのポインタを指定します。取得しない場合は、NULL を指定します。以下の値の組み合わせの値が取得できます。

CW_FS_BOLD	(1)	太字
CW_FS_ITALIC	(2)	斜体
CW_FS_UNDERLINE	(4)	下線

nFontSize

フォントサイズを格納する INT 型へのポインタを指定します。取得しない場合は、NULL を指定します。フォントサイズの単位は 1/10 ポイント単位です。

crFontColor

フォント色を格納する COLORREF 型へのポインタを指定します。取得しない場合は、NULL を指定します。

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetFont , cwSetFontByGrpIdx , cwGetFontByGrpIdx ,
cwSetTextAlign , cwSetTextAlignByGrpIdx ,cwGetTextAlign ,
cwGetTextAlignByGrpIdx , cwSetSpace , cwSetSpaceByGrpIdx ,
cwGetSpace , cwGetSpaceByGrpIdx , cwSetTextEscapement ,
cwSetTextEscapementByGrpIdx , cwGetTextEscapement ,
cwGetTextEscapementByGrpIdx , cwGetTextMetrics , cwGetTextMetricsByGrpIdx ,
cwSetWordwrap , cwGetWordwrap , cwSetWordwrapByGrpIdx , cwGetWordwrapByGrpIdx

cwSetMultiLine

テキスト・アイテムのマルチラインモードを設定します。

C/C++

```
BOOL cwSetMultiLine(  
    HCWREP hCwRep,          //レポートハンドル  
    LPCSTR lpGrpName,      //グループ名  
    LPCSTR lpItemName,     //アイテム名  
    BOOL bMulti            //マルチラインモード  
);
```

VB

```
Declare Function cwSetMultiLine Lib "cwRepEng.dll" (ByVal hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As String, ByVal bMulti As Long) As Long
```

Delphi

```
function cwSetMultiLine (hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; bMulti: Integer ): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

bMulti

テキストアイテムをマルチラインモードにするか否かを設定します。以下の値を指定します。

0 : マルチラインモード無効 (単行表示)

0 以外 : マルチラインモード (複数行表示)

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwGetMultiLine

cwGetMultiLine

テキスト・アイテムのマルチラインモードを取得します。

C/C++

```
BOOL cwGetMultiLine(  
    HCWREP    hCwRep,        //レポートハンドル  
    LPCSTR    lpGrpName,     //グループ名  
    LPCSTR    lpItemName,   //アイテム名  
    BOOL      *bMulti       //マルチラインモード  
);
```

VB

```
Declare Function cwGetMultiLine Lib "cwRepEng.dll" (ByVal hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As String, bMulti As Long) As Long
```

Delphi

```
function cwGetMultiLine (hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; bMulti: Pointer ): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

bMulti

テキストアイテムをマルチラインモードにするか否かの値を格納する BOOL 型のポインタを設定します。以下の値が取得できます。

0 : マルチラインモード無効 (単行表示)

0 以外 : マルチラインモード (複数行表示)

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetMultiLine

cwSetTextAlign

テキスト・アイテムの文字の寄せを設定します。

C/C++

```
BOOL cwSetTextAlign(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,    //アイテム名  
    INT *nHorzAlign,      //文字の寄せ  
    INT *nVertAlign       //行の寄せ  
);
```

VB

```
Declare Function cwSetTextAlign Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String,  
nHorzAlign As Long, nVertAlign As Long) As Long
```

Delphi

```
function cwSetTextAlign(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; nHorzAlign: Pointer; nVertAlign: Pointer ): Integer;  
stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

nHorzAlign

文字の寄せ設定を格納している INT 型へのポインタを指定します。設定しない場合は、NULL を指定します。以下のいずれかの値を指定します。

CW_ALR_LEFT	(0)	左寄せ (上寄せ)
CW_ALR_CENTER	(1)	中央寄せ
CW_ALR_RIGHT	(2)	右寄せ (下寄せ)
CW_ALR_EQUAL	(3)	均等配置

括弧内は縦書きの場合。

nVertAlign

行の寄せ設定を格納している INT 型へのポインタを指定します。設定しない場合は、NULL を指定します。以下のいずれかの値を指定します。

CW_ATB_TOP	(0)	上寄せ (右寄せ)
CW_ATB_CENTER	(1)	中央寄せ
CW_ATB_BOTTOM	(2)	下寄せ (左寄せ)

CW_ATB_EQUAL (3) 均等配置
括弧内は縦書きの場合。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetFont , cwSetFontByGrpIdx , cwGetFont , cwGetFontByGrpIdx ,
cwSetTextAlignByGrpIdx , cwGetTextAlign , cwGetTextAlignByGrpIdx ,
cwSetSpace , cwSetSpaceByGrpIdx , cwGetSpace , cwGetSpaceByGrpIdx ,
cwSetTextEscapement , cwSetTextEscapementByGrpIdx ,
cwGetTextEscapement , cwGetTextEscapementByGrpIdx , cwGetTextMetrics ,
cwGetTextMetricsByGrpIdx , cwSetWordwrap , cwGetWordwrap ,
cwSetWordwrapByGrpIdx , cwGetWordwrapByGrpIdx

cwGetTextAlign

テキスト・アイテムの文字の寄せを設定します。

C/C++

```
BOOL cwGetTextAlign(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,    //アイテム名  
    INT *nHorzAlign,      //文字の寄せ  
    INT *nVertAlign       //行の寄せ  
);
```

VB

```
Declare Function cwGetTextAlign Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String,  
nHorzAlign As Long, nVertAlign As Long) As Long
```

Delphi

```
function cwGetTextAlign(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; nHorzAlign: Pointer; nVertAlign: Pointer ): Integer;  
stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

nHorzAlign

文字の寄せ設定を格納する INT 型へのポインタを指定します。取得しない場合は、NULL を指定します。

CW_ALR_LEFT	(0)	左寄せ (上寄せ)
CW_ALR_CENTER	(1)	中央寄せ
CW_ALR_RIGHT	(2)	右寄せ (下寄せ)
CW_ALR_EQUAL	(3)	均等配置

括弧内は縦書きの場合。

nVertAlign

行の寄せ設定を格納する INT 型へのポインタを指定します。取得しない場合は、NULL を指定します。

CW_ATB_TOP	(0)	上寄せ (右寄せ)
CW_ATB_CENTER	(1)	中央寄せ
CW_ATB_BOTTOM	(2)	下寄せ (左寄せ)

CW_ATB_EQUAL (3) 均等配置
括弧内は縦書きの場合。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetFont` , `cwSetFontByGrpIdx` , `cwGetFont` , `cwGetFontByGrpIdx` ,
`cwSetTextAlign` , `cwSetTextAlignByGrpIdx` , `cwGetTextAlignByGrpIdx` ,
`cwSetSpace` , `cwSetSpaceByGrpIdx` , `cwGetSpace` , `cwGetSpaceByGrpIdx` ,
`cwSetTextEscapement` , `cwSetTextEscapementByGrpIdx` ,
`cwGetTextEscapement` , `cwGetTextEscapementByGrpIdx` , `cwGetTextMetrics` ,
`cwGetTextMetricsByGrpIdx` , `cwSetWordwrap` , `cwGetWordwrap` ,
`cwSetWordwrapByGrpIdx` , `cwGetWordwrapByGrpIdx`

cwSetSpace

テキスト・アイテムの文字・行の間隔を設定します。

C/C++

```
BOOL cwSetSpace(  
    HCWREP hCwRep,          //レポートハンドル  
    LPCSTR lpGrpName,      //グループ名  
    LPCSTR lpItemName,     //アイテム名  
    INT *nCharSpace,       //文字の寄せ  
    INT *nLineSpace        //行の寄せ  
);
```

VB

```
Declare Function cwSetSpace Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String,  
nCharSpace As Long, nLineSpace As Long) As Long
```

Delphi

```
function cwSetSpace(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; nCharSpace: Pointer; nLineSpace: Pointer): Integer;  
stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

nCharSpace

文字の間隔を格納している INT 型へのポインタを指定します。設定しない場合は、NULL を指定します。設定する値の単位はレポートの長さの単位により異なります。単位がミリメートルの場合、1/100 ミリメートルを基本単位とします。インチの場合は、1 / 1000 インチを基本単位とします。

nLineSpace

行の間隔を格納している INT 型へのポインタを指定します。設定しない場合は、NULL を指定します。設定する値の単位はレポートの長さの単位により異なります。単位がミリメートルの場合、1/100 ミリメートルを基本単位とします。インチの場合は、1 / 1000 インチを基本単位とします。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetFont` , `cwSetFontByGrpIdx` , `cwGetFont` , `cwGetFontByGrpIdx` ,
`cwSetTextAlign` , `cwSetTextAlignByGrpIdx` , `cwGetTextAlign` ,
`cwGetTextAlignByGrpIdx` , `cwSetSpaceByGrpIdx` , `cwGetSpace` ,
`cwGetSpaceByGrpIdx` , `cwSetTextEscapement` ,
`cwSetTextEscapementByGrpIdx` , `cwGetTextEscapement` ,
`cwGetTextEscapementByGrpIdx` , `cwGetTextMetrics` , `cwGetTextMetricsByGrpIdx` ,
`cwSetWordwrap` , `cwGetWordwrap` , `cwSetWordwrapByGrpIdx` , `cwGetWordwrapByGrpIdx`

cwGetSpace

テキスト・アイテムの文字・行の間隔の設定を取得します。

C/C++

```
BOOL cwGetSpace(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,    //アイテム名  
    INT *nCharSpace,      //文字の寄せ  
    INT *nLineSpace       //行の寄せ  
);
```

VB

```
Declare Function cwGetSpace Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String,  
nCharSpace As Long, nLineSpace As Long) As Long
```

Delphi

```
function cwGetSpace(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; nCharSpace: Pointer; nLineSpace: Pointer): Integer;  
stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

nCharSpace

文字の間隔を格納する INT 型へのポインタを指定します。取得しない場合は、NULL を指定します。取得する値の単位はレポートの長さの単位により異なります。単位がミリメートルの場合、1/100 ミリメートルを基本単位とします。インチの場合は、1 / 1000 インチを基本単位とします。

nLineSpace

行の間隔を格納する INT 型へのポインタを指定します。取得しない場合は、NULL を指定します。取得する値の単位はレポートの長さの単位により異なります。単位がミリメートルの場合、1/100 ミリメートルを基本単位とします。インチの場合は、1 / 1000 インチを基本単位とします。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetFont , cwSetFontByGrpIdx , cwGetFont , cwGetFontByGrpIdx ,
cwSetTextAlign , cwSetTextAlignByGrpIdx , cwGetTextAlign ,
cwGetTextAlignByGrpIdx , cwSetSpace ,cwSetSpaceByGrpIdx ,
cwGetSpaceByGrpIdx , cwSetTextEscapement ,cwSetTextEscapementByGrpIdx ,
cwGetTextEscapement ,cwGetTextEscapementByGrpIdx, cwGetTextMetrics,
cwGetTextMetricsByGrpIdx, cwSetWordwrap, cwGetWordwrap,
cwSetWordwrapByGrpIdx, cwGetWordwrapByGrpIdx

cwSetTextEscapement

テキスト・アイテムの文字の横書き・縦書きを設定します。

C/C++

```
BOOL cwSetTextEscapement(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,    //アイテム名  
    INT     nEscMode      //横書き・縦書きの設定  
);
```

VB

```
Declare Function cwSetTextEscapement Lib "cwRepEng.dll"(ByVal  
hCwRep As Long,ByVal lpGrpName As String,ByVal lpItemName As  
String,ByVal nEscMode As Long)As Long
```

Delphi

```
function cwSetTextEscapement(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; nEscMode: Integer ): Integer;stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

nEscMode

横書き・縦書きの値を指定します。以下のいずれかの値を指定します。

0:横書き 1:縦書き

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetFont , cwSetFontByGrpIdx , cwGetFont , cwGetFontByGrpIdx ,
cwSetTextAlign , cwSetTextAlignByGrpIdx , cwGetTextAlign ,
cwGetTextAlignByGrpIdx , cwSetSpace , cwSetSpaceByGrpIdx , cwGetSpace ,
cwGetSpaceByGrpIdx , cwSetTextEscapementByGrpIdx , cwGetTextEscapement ,
cwGetTextEscapementByGrpIdx , cwGetTextMetrics , cwGetTextMetricsByGrpIdx ,
cwSetWordwrap , cwGetWordwrap , cwSetWordwrapByGrpIdx , cwGetWordwrapByGrpIdx

cwGetTextEscapement

テキスト・アイテムの文字の横書き・縦書きの設定を取得します。

C/C++

```
BOOL cwGetTextEscapement(  
HCWREP hCwRep,      //レポートハンドル  
LPCSTR lpGrpName,   //グループ名  
LPCSTR lpItemName,  //アイテム名  
INT *nEscMode       //横書き・縦書きの設定  
);
```

VB

```
Declare Function cwGetTextEscapement Lib "cwRepEng.dll"(ByVal  
hCwRep As Long,ByVal lpGrpName As String,ByVal lpItemName As  
String,nEscMode As Long )As Long
```

Delphi

```
function cwGetTextEscapement(hCwRep:Integer;lpGrpName:PChar;  
lpItemName:PChar;nEscMode:Pointer ):Integer;stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

nEscMode

横書き・縦書きの設定値を格納する INT 型へのポインタを指定します。以下のいずれかの値が取得できます。

0:横書き 1:縦書き

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetFont , cwSetFontByGrpIdx , cwGetFont , cwGetFontByGrpIdx ,
cwSetTextAlign , cwSetTextAlignByGrpIdx , cwGetTextAlign ,
cwGetTextAlignByGrpIdx , cwSetSpace , cwSetSpaceByGrpIdx , cwGetSpace ,
cwGetSpaceByGrpIdx , cwSetTextEscapement , cwSetTextEscapementByGrpIdx ,
cwGetTextEscapementByGrpIdx , cwGetTextMetrics , cwGetTextMetricsByGrpIdx ,

`cwSetWordwrap`, `cwGetWordwrap`, `cwSetWordwrapByGrpIdx`, `cwGetWordwrapByGrpIdx`

cwGetTextMetrics

テキストアイテムの印刷する文字列の高さ・幅を取得します。

C/C++

```
BOOL cwGetTextMetrics(  
    HCWREP hCwRep,           //レポートハンドル  
    LPCSTR lpGrpName,       //グループ名  
    LPCSTR lpItemName,     //アイテム名  
    DOUBLE *dbWidth,       //文字列の高さ  
    DOUBLE *dbHeight       //文字列の幅  
);
```

VB

```
Public Declare Function cwGetTextMetrics Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, dbWidth As Double, dbHeight As Double) As Long
```

Delphi

```
function cwGetTextMetrics(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; dbWidth: Pointer; dbHeight: Pointer ): Integer;  
stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

dbWidth

文字列の幅を取得する DOUBLE 型のポインタを指定します。長さの単位は、レポートの長さの単位に従い、ミリメートル/インチとなります。文字列の幅は、指定した文字間隔が含まれたものが返ります。

dbHeight

文字列の高さを取得する DOUBLE 型のポインタを指定します。長さの単位は、レポートの長さの単位に従い、ミリメートル/インチとなります。指定した行の間隔は含まれません。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetFont , cwSetFontByGrpIdx , cwGetFont , cwGetFontByGrpIdx ,
cwSetTextAlign , cwSetTextAlignByGrpIdx , cwGetTextAlign ,
cwGetTextAlignByGrpIdx , cwSetSpace , cwSetSpaceByGrpIdx ,cwGetSpace ,
cwGetSpaceByGrpIdx , cwSetTextEscapement ,cwSetTextEscapementByGrpIdx ,
cwGetTextEscapementByGrpIdx, cwSetWordwrap, cwGetWordwrap,
cwSetWordwrapByGrpIdx, cwGetWordwrapByGrpIdx

cwSetWordwrap

テキスト・アイテムのテキストの折り返しを設定します。

C/C++

```
BOOL cwSetWordwrap(  
    HCWREP hCwRep,           //レポートハンドル  
    LPCSTR IpGrpName,       //グループ名  
    LPCSTR IpItemName,     //アイテム名  
    BOOL bWordwrap         //折り返しの設定  
);
```

VB

```
Declare Function cwSetWordwrap Lib "cwRepEng.dll" (ByVal hCwRep As Long, ByVal IpGrpName As String, ByVal IpItemName As String, ByVal bWordwrap As Long) As Long
```

Delphi

```
function cwSetWordwrap(hCwRep: Integer; IpGrpName: PChar;  
    IpItemName: PChar; bWordwrap: Integer): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

IpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

IpItemName

アイテム名へのポインタを指定します。

bWordwrap

キストの折り返しの設定を指定します。以下のいずれかの値を指定します。

0 : :テキストを折り返さない

0 以外 : :テキストを折り返す

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetFont , cwSetFontByGrpIdx , cwGetFont , cwGetFontByGrpIdx ,
cwSetTextAlign , cwSetTextAlignByGrpIdx , cwGetTextAlign ,
cwGetTextAlignByGrpIdx , cwSetSpace , cwSetSpaceByGrpIdx , cwGetSpace ,
cwGetSpaceByGrpIdx , cwSetTextEscapement , cwSetTextEscapementByGrpIdx ,
cwGetTextEscapementByGrpIdx , cwGetWordwrap , cwSetWordwrapByGrpIdx ,

cwGetWordwrapByGrpIdx

cwGetWordwrap

テキスト・アイテムのテキストの折り返しの設定を取得します。

C/C++

```
BOOL cwGetWordwrap(  
    HCWREP hCwRep,           //レポートハンドル  
    LPCSTR IpGrpName,       //グループ名  
    LPCSTR IpItemName,     //アイテム名  
    BOOL *bWordwrap        //折り返しの設定  
);
```

VB

```
Declare Function cwGetWordwrap Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal IpGrpName As String, ByVal IpItemName As String,  
bWordwrap As Long )As Long
```

Delphi

```
function cwGetWordwrap(hCwRep: Integer; IpGrpName: PChar;  
IpItemName: PChar; bWordwrap: Pointer ): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

IpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

IpItemName

アイテム名へのポインタを指定します。

bWordwrap

テキストの折り返しの設定を取得する BOOL 型のポインタを指定します。以下のいずれかの値を取得できます。

0 : :テキストを折り返さない

0 以外 : :テキストを折り返す

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetFont , cwSetFontByGrpIdx , cwGetFont , cwGetFontByGrpIdx ,
cwSetTextAlign , cwSetTextAlignByGrpIdx , cwGetTextAlign ,
cwGetTextAlignByGrpIdx , cwSetSpace , cwSetSpaceByGrpIdx , cwGetSpace ,
cwGetSpaceByGrpIdx , cwSetTextEscapement , cwSetTextEscapementByGrpIdx ,

cwGetTextEscapementByGrpIdx, cwSetWordwrap, cwSetWordwrapByGrpIdx,
cwGetWordwrapByGrpIdx

cwSetBarcodeProp

バーコード・アイテムの詳細な設定をします。

C/C++

```
BOOL WINAPI cwSetBarcodeProp(  
    HCWREP      hCwRep,          //レポートハンドル  
    LPCSTR      lpGrpName,       //グループ名  
    LPCSTR      lpItemName,      //アイテム名  
    CW_BARCODEDEF *lpBarcodeDef //バーコードプロパティ  
);
```

VB

```
Declare Function cwSetBarcodeProp Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal lpGrpName As String, ByVal lpItemName As String,  
lpBarcodeDef As CW_BARCODEDEF) As Long
```

Delphi

```
function cwSetBarcodeProp(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; lpBarcodeDef: Pointer): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

lpBarcodeDef

バーコードプロパティ・データブロックへのポインタを指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetText , cwSetTextByGrpIdx , cwGetText , cwGetTextByGrpIdx ,
cwSetBarcodePropByGrpIdx , cwGetBarcodeProp ,
cwGetBarcodePropByGrpIdx

cwGetBarcodeProp

バーコード・アイテムの詳細な設定を取得します。

C/C++

```
BOOL WINAPI cwGetBarcodeProp(  
    HCWREP        hCwRep,        //レポートハンドル  
    LPCSTR        lpGrpName,     //グループ名  
    LPCSTR        lpItemName,    //アイテム名  
    CW_BARCODEDEF *lpBarcodeDef //バーコードプロパティ  
);
```

VB

```
Declare Function cwGetBarcodeProp Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal lpGrpName As String, ByVal lpItemName As String,  
lpBarcodeDef As CW_BARCODEDEF) As Long
```

Delphi

```
function cwGetBarcodeProp(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; lpBarcodeDef: Pointer): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

lpBarcodeDef

取得するバーコードプロパティ・データブロックを格納するバッファのポインタを指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetText , cwSetTextByGrpIdx , cwGetText , cwGetTextByGrpIdx ,
cwSetBarcodeProp , cwSetBarcodePropByGrpIdx ,
cwGetBarcodePropByGrpIdx

cwSetImgStretch

イメージ・アイテムの領域に収まるようにイメージを拡大・縮小するかを設定をします。

C/C++

```
BOOL cwSetImgStretch(  
    HCWREP hCwRep,      //レポートハンドル  
    LPCSTR lpGrpName,   //グループ名  
    LPCSTR lpItemName,  //アイテム名  
    BOOL bImgStretch    //ストレッチ  
);
```

VB

```
Declare Function cwSetImgStretch Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal lpGrpName As String, ByVal lpItemName As String, ByVal  
bImgStretch As Long) As Long
```

Delphi

```
function cwSetImgStretch(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; bImgStretch: Integer): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

bStretch

イメージアイテムの領域に納まるようにイメージを拡大縮小するかを指定します。

0 : 変形しない

0 以外 : 収まるようにする

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetImagePath , cwSetImagePathByGrpIdx , cwSetImageDIB ,
cwSetImageDIBByBrpIdx , cwSetImgStretchByGrpIdx ,
cwGetImgStretch , cwGetImgStretchByGrpIdx

cwGetImgStretch

イメージ・アイテムの領域に収まるようにイメージを拡大・縮小するかの設定を取得します。

C/C++

```
BOOL cwGetImgStretch(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,    //アイテム名  
    BOOL *bImgStretch    //ストレッチ  
);
```

VB

```
Declare Function cwGetImgStretch Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal lpGrpName As String, ByVal lpItemName As String,  
bImgStretch As Long) As Long
```

Delphi

```
function cwGetImgStretch(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; bImgStretch: Pointer): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

bStretch

イメージアイテムの領域に納まるようにイメージを拡大縮小するかを指定を取得する BOOL 型のポインタを指定します。以下のいずれかの値が取得できます。

0 : 変形しない

0 以外 : 収まるようにする

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetImagePath , cwSetImagePathByGrpIdx , cwSetImageDIB ,
cwSetImageDIBByBrpIdx , cwSetImgStretch , cwSetImgStretchByGrpIdx ,
cwGetImgStretchByGrpIdx

cwSetVisible

アイテムを表示するか否かを設定をします。

C/C++

```
BOOL cwSetVisible(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,    //アイテム名  
    BOOL   bVisible       //表示・非表示  
);
```

VB

```
Declare Function cwSetVisible Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String, ByVal  
bVisible As Long) As Long
```

Delphi

```
function cwSetVisible(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; bVisible: Integer): Integer; stdcall; external  
'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

bVisible

アイテムを表示・非表示するかを指定します。以下のいずれかの値を指定します。

0 : 表示しない

0 以外 : 表示する

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetBound , cwSetBoundByGrpIdx , cwGetBound , cwGetBoundByGrpIdx ,
cwSetVisibleByGrpIdx , cwGetVisible , cwGetVisibleByGrpIdx ,
cwSetForeground , cwSetForegroundByGrpIdx

cwGetVisible

アイテムを表示するか否かの設定を取得します。

C/C++

```
BOOL cwGetVisible(  
    HCWREP hCwRep,      //レポートハンドル  
    LPCSTR lpGrpName,   //グループ名  
    LPCSTR lpItemName,  //アイテム名  
    BOOL *bVisible     //表示・非表示  
);
```

VB

```
Declare Function cwGetVisible Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String, bVisible  
As Long) As Long
```

Delphi

```
function cwGetVisible(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; bVisible: Pointer): Integer; stdcall; external  
'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

bVisible

アイテムを表示・非表示するかの設定を格納する BOOL 型のポインタを指定します。

0 : 表示しない

0 以外 : 表示する

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetBound , cwSetBoundByGrpIdx , cwGetBound , cwGetBoundByGrpIdx ,
cwSetVisible , cwSetVisibleByGrpIdx , cwGetVisibleByGrpIdx ,
cwSetForeground , cwSetForegroundByGrpIdx

cwSetGrpVisible

グループを表示するか否かの設定を取得します。

C/C++

```
BOOL cwSetGrpVisible(  
    HCWREP hCwRep,      //レポートハンドル  
    LPCSTR lpGrpName,   //グループ名  
    BOOL bVisible,     //表示・非表示  
    INT nItemFlags     //設定したいアイテムの種類  
);
```

VB

```
Declare Function cwSetGrpVisible Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal bVisible As  
    Long, ByVal nItemFlags) As Long
```

Delphi

```
function cwSetGrpVisible (hCwRep:Integer;lpGrpName:PChar;  
    bVisible:Integer; nItemFlags: Integer):Integer;stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。

bVisible

グループを表示・非表示するかを指定します。以下のいずれかの値を指定します。

0 : 表示しない

0以外 : 表示する

nItemFlags

Visible プロパティを設定するグループ内のアイテムの種類を指定します。以下の値の和を指定します。グループ内全てのアイテムを対象とする場合は、-1を指定します。

1 : テキスト アイテムに設定する。

2 : イメージ アイテムに設定する。

4 : バーコード アイテムに設定する。

8 : 線 (Line) アイテムに設定する。

16 : 枠 (Box) アイテムの設定する。

32 : 円 (Ellipse) アイテムに設定する。

例) グループ内のテキスト, 線 (Line) アイテムの Visible プロパティを設定したい場合は、nItemFlags に 9 (1 + 8) を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetBound` , `cwSetBoundByGrpIdx` , `cwGetBound` , `cwGetBoundByGrpIdx` ,
`cwSetVisible` , `cwGetVisible`, `cwSetVisibleByGrpIdx` , `cwGetVisibleByGrpIdx` ,
`cwSetForeground` , `cwSetForegoundByGrpIdx`

cwSetForeground

アイテムを前面・背面に移動させます。

C/C++

```
BOOL cwSetForeground(  
    HCWREP hCwRep,      //レポートハンドル  
    LPCSTR lpGrpName,   //グループ名  
    LPCSTR lpItemName,  //アイテム名  
    BOOL   bFore        //前面・背面移動  
);
```

VB

```
Declare Function cwSetForeground Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal lpGrpName As String, ByVal lpItemName As String, ByVal  
bFore As Long) As Long
```

Delphi

```
function cwSetForeground(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; bFore: Integer): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

bFore

アイテムを前面・背面へ移動するかを指定します。

0 : 背面に移動

0 以外 : 前面に移動

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetBound , cwSetBoundByGrpIdx , cwGetBound , cwGetBoundByGrpIdx ,
cwSetVisible , cwSetVisibleByGrpIdx , cwGetVisible ,
cwGetVisibleByGrpIdx , cwSetForegroundByGrpIdx

cwGetItemType

アイテムの種別を取得します。

C/C++

```
INT cwGetItemType(  
    HCWREP hCwRep,      //レポートハンドル  
    LPCSTR IpGrpName,  //グループ名  
    LPCSTR IpItemName, //アイテム名  
);
```

VB

```
Declare Function cwGetItemType Lib "cwRepEng.dll" (ByVal hCwRep As Long, ByVal IpGrpName As String, ByVal IpItemName As String) As Long
```

Delphi

```
function cwGetItemType(hCwRep: Integer; IpGrpName: PChar;  
    IpItemName: PChar): Integer; stdcall; external 'cwRepEng.dll'
```

パラメータ

hCwRep

レポートのハンドルを指定します。

IpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

IpItemName

アイテム名へのポインタを指定します。

戻り値

関数が成功すると、0以外のアイテムの種別を示す値が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

アイテムの種別

CW_ITEMTYPE_TEXT	(1)	: テキスト
CW_ITEMTYPE_IMAGE	(2)	: イメージ
CW_ITEMTYPE_BARCODE	(3)	: バーコード
CW_ITEMTYPE_LINE	(4)	: 線(Line)
CW_ITEMTYPE_BOX	(5)	: 枠(Box)
CW_ITEMTYPE_ELLIPSE	(6)	: 円(Ellipse)

cwRenameItem

アイテムのグループ名及び名前を変えます。

C/C++

```
BOOL cwRenameItem(  
    HCWREP hCwRep,          //レポートハンドル  
    LPCSTR lpGrpName,      //グループ名  
    LPCSTR lpItemName,     //アイテム名  
    LPCSTR lpNewGrpName,   //新しいグループ名  
    LPCSTR lpNewItemName  //新しいアイテム名  
);
```

VB

```
Declare Function cwRenameItem Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String, ByVal  
lpNewGrpName As String, ByVal lpNewItemName As String) As Long
```

Delphi

```
function cwRenameItem(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; lpNewGrpName: PChar; lpNewItemName: PChar ):  
Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

lpNewGrpName

変更後のグループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpNewItemName

変更後のアイテム名へのポインタを指定します。

戻り値

関数が成功すると、0以外の値が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

cwSetFixed

アイテムを固定項目とするか否かを設定をします。

C/C++

```
BOOL cwSetFixed(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,   //アイテム名  
    BOOL   bFixed        //固定項目とする・しない  
);
```

VB

```
Declare Function cwSetFixed Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String, ByVal  
bFixed As Long) As Long
```

Delphi

```
function cwSetFixed(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; bFixed: Integer): Integer; stdcall; external  
'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

bFixed

アイテムを固定項目とするか否かを指定します。以下のいずれかの値を指定します。

0 : 固定項目としない

0 以外 : 固定項目とする

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwGetFixed , cwSetFixedByGrpIdx , cwGetFixedByGrpIdx

cwGetFixed

アイテムを固定項目とするか否かの設定を取得します。

C/C++

```
BOOL cwGetFixed(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,   //アイテム名  
    BOOL *bFixed         //固定項目とする・しない  
);
```

VB

```
Declare Function cwGetFixed Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String, bFixed  
As Long) As Long
```

Delphi

```
function cwGetFixed(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; bFixed: Pointer): Integer; stdcall; external  
'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

bFixed

アイテムを固定項目とするか否かの設定を格納する BOOL 型のポインタを指定します。

0 : 固定項目としない

0 以外 : 固定項目とする

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetFixed , cwSetFixedByGrpIdx , cwGetFixedByGrpIdx ,

2.2.4 アイテムリピートモード関数

cwStartRepeat

アイテムまたはグループをリピートモードに設定します。
リピートモードに設定されたアイテムまたはグループは、cwPrintReport 関数で印刷されると、リピートモードが解除されます。

リピートモードとは、指定方向・指定間隔で、そのアイテムまたはグループの複製を、自動で作成する機能です。これにより、一覧表などの印刷を簡単に作ることができます。

C/C++

```
BOOL cwStartRepeat(  
    HCWREP hCwRep,           //レポートのハンドル  
    LPCSTR IpGrpName,       //グループ名  
    LPCSTR IpItemName,     //アイテム名  
    BOOL bVertical,        //複製を作る方向  
    INT nInterval,         //複製を作る間隔  
);
```

VB

```
Declare Function cwStartRepeat Lib "cwRepEng.dll" (ByVal hCwRep As  
Long,ByVal IpGrpName As String,ByVal IpItemName As String,ByVal  
bVertical As Long,Byval nInterval As Long )As Long
```

Delphi

```
function  
cwStartRepeat(hCwRep: Integer; IpGrpName: PChar; IpItemName: PChar;  
bVertical : Integer;nInterval: Integer ): Integer;stdcall;external  
'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

IpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

IpItemName

アイテム名へのポインタを指定します。

グループ全体をリピートモード(グループリピートモード)にする場合は、NULL または "" を指定します。このグループリピートモードはアイテムリピートモー

ドとは異なり、cwDupGrp 関数で複製を作ります。

bVertical

複製を作る方向

TRUE: 縦下方向に複製を作成する。

FALSE: 横右方向に複製を作成する。

nInterval

複製を作る間隔を指定します。通常は「自動的に次のアイテムを複製する」値の -1 を指定します。

レポートの長さの単位に従います。単位がミリメートルの場合、1/100 ミリメートルを基本単位とします。インチの場合は、1 / 1000 インチを基本単位とします。

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwCheckDupBound, cwSetText, cwSetIntToStr, cwSetDoubleToStr, cwSetImagePath, cwSetImageDIB, cwSetDIBits, cwDupBoxLine, cwDupGrp

cwDupBoxLine

アイテムリピートモードに設定された線・枠 (Box・Line) アイテムの複製を cwStartRepeat 関数で指定された方向、間隔で作成します。

指定するアイテムは、cwStartRepeat 関数でリピートモードに設定されている必要があります。

C/C++

```
BOOL cwDupBoxLine(  
    HCWREP hCwRep,          //レポートのハンドル  
    LPCSTR lpGrpName,      //グループ名  
    LPCSTR lpItemName     //アイテム名  
);
```

VB

```
Declare Function cwDupBoxLine Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String) As Long
```

Delphi

```
function cwDupBoxLine(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetText, cwSetIntToStr, cwSetDoubleToStr, cwSetImagePath, cwSetImageDIB,
cwSetDIBits, cwDupGrp

cwDupGrp

グループリピートモードに設定されたグループの複製を cwStartRepeat 関数で指定された方向、間隔で作成します。

指定するグループは、cwStartRepeat 関数でグループリピートモードに設定されている必要があります。

C/C++

```
BOOL cwDupGrp(  
    HCWREP hCwRep,    //レポートのハンドル  
    LPCSTR IpGrpName //グループ名  
);
```

VB

```
Declare Function cwDupGrp Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal IpGrpName As String) As Long¥
```

Delphi

```
function cwDupGrp(hCwRep: Integer; IpGrpName: PChar ): Integer;  
stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

IpGrpName

グループ名へのポインタを指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetText, cwSetIntToStr, cwSetDoubleToStr, cwSetImagePath, cwSetImageDIB,
cwSetDIBits, cwDupBoxLine

cwCheckDupBound

次に複製されるグループまたはアイテムがレポートの印刷範囲内に収まるか否かをチェックします。収まらない場合、エラー（エラー番号5を内部に記録）を返し、レポートの印刷可能範囲をパラメータに戻します。

C/C++

```
BOOL cwCheckDupBound(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,   //アイテム名  
    INT *nLeft,          //左位置  
    INT *nTop,           //上位置  
    INT *nWidth,         //幅  
    INT *nHeight         //高さ  
);
```

VB

```
Declare Function cwCheckDupBound Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal lpGrpName As String, ByVal lpItemName As String, nLeft  
As Long, nTop As Long, nWidth As Long, nHeight As Long) As Long
```

Delphi

```
function cwCheckDupBound(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; nLeft: Pointer; nTop: Pointer; nWidth: Pointer;  
nHeight: Pointer): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULLを指定します。

lpItemName

アイテム名へのポインタを指定します。

nLeft

左位置の値を格納する INT 型変数へのポインタ。線アイテムの場合は、線の始端の左右方向の位置を格納する INT 型変数へのポインタ。取得しない場合は、NULLを指定します。

nTop

位置の値を格納する INT 型変数へのポインタ。線アイテムの場合は、線の始端の上下方向の位置を格納する INT 型変数へのポインタ。取得しない場合は、NULLを指定します。

nWidth

幅の値を格納する INT 型変数へのポインタ。線アイテムの場合は、線の終端の左

右方向の位置を格納する INT 型変数へのポインタ。取得しない場合は、NULL を指定します。

nHeight

高さの値を格納する INT 型変数へのポインタ。線アイテムの場合は、線の終端の上下方向の位置を格納する INT 型変数へのポインタ。取得しない場合は、NULL を指定します。

位置・大きさの値は、レポートの長さの単位により異なります。単位がミリメートルの場合、1/100 ミリメートルを基本単位とします。インチの場合は、1 / 1000 インチを基本単位とします。

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。取得したエラー情報がエラー番号 5 の場合、次に複製されるグループまたはアイテムがレポートの印刷範囲内に作成できないことを意味します。この場合、パラメータには、印刷可能範囲が戻されます。

参照

cwSetText, cwSetIntToStr, cwSetDoubleToStr, cwSetImagePath, cwSetImageDIB, cwSetDIBits, cwDupBoxLine, cwDupGrp

2.2.5 アイテム一覧取得関数

cwGetItemList

アイテム名の一覧を取得します。VB の場合は、cwGetFisrtItem/cwGetNextItem 関数を使用してください。

C/C++

```
BOOL WINAPI cwGetItemList(  
    HCWREP hCwRep,           //レポートハンドル  
    CWREPLISTPROC lpListProc //コールバック関数  
);
```

VB

```
Declare Function cwGetItemList Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpListProc As Long) As Long
```

Delphi

```
function cwGetItemList(hCwRep: Integer; lpListProc: Pointer):  
Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpListProc

一覧取得を処理するコールバック関数

コールバック関数の型

```
INT CALLBACK CWREPLISTPROC(  
    LPCSTR lpGrpName, //グループ名  
    LPCSTR lpItemName, //アイテム名  
    INT nKind //アイテムの種類  
);
```

コールバック関数のパラメータ

lpGrpName

アイテムのグループ名を格納した文字列へのポインタが渡されます。文字列ポインタが示すバッファの内容はコールバック関数内でのみ有効となります。必ず文字列は別のバッファにコピーしてから参照してください。

lpItemName

アイテム名を格納した文字列へのポインタが渡されます。文字列ポインタが示すバッファの内容はコールバック関数内でのみ有効となります。必ず文字列は別のバッファにコピーしてから参照してください。

nKind

アイテムの種類が渡されます。種類は以下のとおりです。

CW_ITEMTYPE_TEXT (1) : テキスト

CW_ITEMTYPE_IMAGE (2) : イメージ
CW_ITEMTYPE_BARCODE (3) : バーコード
CW_ITEMTYPE_LINE(4) : 線(Line)
CW_ITEMTYPE_BOX (5) : 枠(Box)
CW_ITEMTYPE_ELLIPSE (6) : 円(Ellipse)

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwGetFirstItem`

cwGetFirstItem

選択されているページ内の先頭アイテムのグループ名・アイテム名・種別を取得します。

C/C++

```
HCWITEM cwGetFirstItem(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,    //アイテム名  
    INT *nKnd             //アイテムの種類  
);
```

VB

```
Declare Function cwGetFirstItem Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String, nKnd As  
Long) As Long
```

Delphi

```
function cwGetFirstItem(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; nKnd: Pointer): Integer; stdcall; external  
'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名を格納するバッファのポインタを指定します。グループ名は最大 64 Byte の長さのバッファを必要とします。

lpItemName

アイテム名を格納するバッファのポインタを指定します。アイテム名は最大 64 Byte の長さのバッファを必要とします。

nKnd

アイテムの種類を格納する INT 型のポインタを指定します。以下のいずれかの値が格納されます。

CW_ITEMTYPE_TEXT (1)	:	テキスト
CW_ITEMTYPE_IMAGE (2)	:	イメージ
CW_ITEMTYPE_BARCODE (3)	:	バーコード
CW_ITEMTYPE_LINE (4)	:	線(Line)
CW_ITEMTYPE_BOX (5)	:	枠(Box)
CW_ITEMTYPE_ELLIPSE (6)	:	円(Ellipse)

戻り値

関数が成功すると、アイテム列挙ハンドルが返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

`cwGetNextItem` , `cwCloseItemHandle`

cwGetNextItem

直前の cwGetFirstItem 関数に続けて、次のアイテムのグループ名・アイテム名・種別を取得します。

C/C++

```
INT cwGetNextItem(  
    HCWREP hCwRep,      //レポートハンドル  
    HCWITEM hCwItem,    //アイテム列挙ハンドル  
    LPCSTR lpGrpName,   //グループ名  
    LPCSTR lpItemName,  //アイテム名  
    INT *nKnd           //アイテムの種類  
);
```

VB

```
Declare Function cwGetNextItem Lib "cwRepEng.dll" (ByVal hCwRep As Long, ByVal hCwItem As Long, ByVal lpGrpName As String, ByVal lpItemName As String, nKnd As Long) As Long
```

Delphi

```
function cwGetNextItem(hCwRep: Integer; hCwItem: Integer;  
    lpGrpName: PChar; lpItemName: PChar; nKnd: Pointer): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名を格納するバッファのポインタを指定します。グループ名は最大 64 Byte の長さのバッファを必要とします。

lpItemName

アイテム名を格納するバッファのポインタを指定します。アイテム名は最大 64 Byte の長さのバッファを必要とします。

nKnd

アイテムの種類を格納する INT 型のポインタを指定します。以下のいずれかの値が格納されます。

CW_ITEMTYPE_TEXT (1)	:	テキスト
CW_ITEMTYPE_IMAGE (2)	:	イメージ
CW_ITEMTYPE_BARCODE (3)	:	バーコード
CW_ITEMTYPE_LINE (4)	:	線(Line)
CW_ITEMTYPE_BOX (5)	:	枠(Box)
CW_ITEMTYPE_ELLIPSE (6)	:	円(Ellipse)

戻り値

アイテムの情報が取得できた場合、0 以外の値が返ります。列挙すべきアイテムが無い場合、0 が返ります。

参照

cwGetFirstItem , cwCloseItemHandle

cwCloseItemHandle

cwGetFirstItem 関数で取得したアイテム列挙ハンドルを開放します。

C/C++

```
BOOL cwCloseItemHandle (  
    HCWREP hCwRep,        //レポートのハンドル  
    HCWITEM hCwItem      //アイテム列挙ハンドル  
);
```

VB

```
Declare Function cwCloseItemHandle Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal hCwItem As Long) As Long
```

Delphi

```
function cwCloseItemHandle(hCwRep: Integer; hCwItem: Integer):  
Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

hCwItem

cwGetFirstItem 関数で取得したアイテム列挙ハンドルを指定します。

戻り値

関数が成功すると、0以外の値が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwGetFirstItem , cwGetNextItem

2.2.6 インデックス付きグループのアイテムプロパティ操作関数

cwSetImagePathByGrpIdx

イメージ・アイテムに指定のイメージファイルからイメージを読み込みます。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwSetImagePathByGrpIdx(  
    HCWREP hCwRep,          //レポートのハンドル  
    LPCSTR lpGrpName,       //グループ名  
    LPCSTR lpItemName,      //アイテム名  
    LPCSTR lpFileName,      //イメージファイルのパス  
    INT nDigit,            //インデックスの桁数  
    INT nIdx                //インデックス  
);
```

VB

```
Declare Function cwSetImagePathByGrpIdx Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, ByVal lpFileName As String, ByVal nDigit As Long, ByVal nIdx  
    As Long) As Long
```

Delphi

```
function cwSetImagePathByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; lpFileName: PChar; nDigit: Integer; nIdx: Integer):  
    Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

lpFileName

イメージファイルへのパスが格納されている文字列のポインタ。(イメージファイルは、TIFF/JPEG/BMP)

http 転送でイメージファイルを開く場合は、イメージファイルへの URL を保持している、NULL で終わる文字列へのポインタを指定します。

ftp 転送でイメージファイルを開く場合は、以下に示す形式でイメージファイルへのパスを保持している、NULL で終わる文字列へのポインタを指定します。

ftp 転送の記述形式

ftp://(ftpサーバー名、又はIPアドレス)/(イメージファイルへのパス)
と記述します。

例) ftpサーバーftp.com-ware.jpで./Sample/Test.jpegを開く場合。
ftp://ftp.com-ware.jp/Sample/Test.jpeg
となります。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、`cwGetLastError`関数を使います。

参照

`cwSetImagePath` , `cwSetImageDIB` , `cwSetImageDIBByBrpIdx` ,
`cwClearItemData` , `cwClearItemDataByGrpIdx` ,
`cwSetInternetParam`

cwSetImageDIBByGrpIdx

イメージをイメージ・アイテムにコピーします。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwSetImageDIBByGrpIdx(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,   //アイテム名  
    HDC     hDc,          //コピー元のデバイスコンテキストのハンドル  
    INT     nXOriginSrc,  //コピー元の X 座標  
    INT     nYOriginSrc,  //コピー元の Y 座標  
    INT     nWidthSrc,    //コピー元の幅  
    INT     nHeightSrc,   //コピー元の高さ  
    INT     nDigit,       //インデックスの桁数  
    INT     nIdx          //インデックス  
);
```

VB

```
Declare Function cwSetImageDIBByGrpIdx Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, ByVal hDc As Long, ByVal nXOriginSrc As Long, ByVal  
    nYOriginSrc As Long, ByVal nWidthSrc As Long, ByVal nHeightSrc As  
    Long, ByVal nDigit As Long, ByVal nIdx As Long) As Long
```

Delphi

```
function cwSetImageDIBByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; hDc: HDC; nXOriginSrc: Integer; nYOriginSrc:  
    Integer; nWidthSrc: Integer; nHeightSrc: Integer; nDigit: Integer;  
    nIdx: Integer): Integer; stdcall; external 'cwRepEng.dll'
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

hDc

コピー元のデバイスコンテキストのハンドル。

XOriginSrc

コピー元の X 座標。

YOriginSrc

コピー元の Y 座標。

nWidth

コピー元の幅。

nHeight

コピー元の高さ。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetImagePath` , `cwSetImagePathByGrpIdx` , `cwSetImageDIB` ,
`cwClearItemData` , `cwClearItemDataByGrpIdx`

cwSetDIBitsByGrpIdx

ビットマップをイメージ・アイテムに設定します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwSetDIBitsByGrpIdx(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR IpGrpName,     //グループ名  
    LPCSTR IpItemName,    //アイテム名  
    LPBITMAPINFO IpBinfo, //ビットマップ情報  
    LPVOID IpBits,        //ビットマップデータ  
    INT nDigit,           //インデックスの桁数  
    INT nIdx              //インデックス  
);
```

VB

```
Public Declare Function cwSetDIBitsByGrpIdx Lib "cwRepEng.dll"  
(ByVal hCwRep As Long,ByVal IpGrpName As String,ByVal  
IpItemName As String,ByVal IpBinfo As Long,ByVal IpBitsData As  
Long,ByVal nDigit As Long,ByVal nIdx As Long)As Long
```

Delphi

```
function cwSetDIBitsByGrpIdx(hCwRep:Integer; IpGrpName:PChar;  
IpItemName:PChar; IpBinfo:Pointer; IpBits:Pointer; nDigit:  
Integer; nIdx:Integer ):Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

IpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

IpItemName

アイテム名へのポインタを指定します。

IpBinfo

DIB の情報を保持している BITMAPINFO 構造体へのポインタを指定します。

IpBits

DIB の色データ（ビットマップのビット配列）を保持しているバイト配列へのポインタを指定します。ビットマップ値の形式は、IpBinfo パラメータが指す BITMAPINFO 構造体の biBitCount メンバに依存します。なお、NULL（0）が指定された場合、パック済み DIB が指定されたものとみなして、処理が行われます。

nDigit

インデックス部分を省いたグループ名に合成（付加）するインデックス部分の桁数を指定します。0 を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetImagePath , cwSetImagePathByGrpIdx , cwSetImageDIB ,
cwClearItemData , cwClearItemDataByGrpIdx

cwSetDPIByGrpIdx

イメージ・アイテムの解像度 (DPI) を設定します。デザイナーで、グループの複製をした時に付けたインデックス (添え字) 部分を変数として扱えるようになっています。

C/C++

```
BOOL cwSetDPIByGrpIdx(  
    HCWREP hCwRep,    //レポートハンドル  
    LPCSTR lpGrpName, //グループ名  
    LPCSTR lpItemName, //アイテム名  
    INT nDpi,        //解像度  
    INT nDigit,      //インデックスの桁数  
    INT nIdx         //インデックス  
);
```

VB

```
Public Declare Function cwSetDPIByGrpIdx Lib "cwRepEng.dll" (ByVal  
hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
String, ByVal nDpi As Long, ByVal nDigit As Long, ByVal nIdx As  
Long) As Long
```

Delphi

```
function function cwSetDPIByGrpIdx(hCwRep: Integer; lpGrpName:  
PChar; lpItemName: PChar; nDpi: Integer; nDigit: Integer; nIdx:  
Integer): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

グループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

アイテム名へのポインタを指定します。

nDpi

イメージの解像度を指定します。イメージアイテムのストレッチが False の場合に有効で、印刷時に解像度に従って適正な大きさにイメージを補正して出力します。0 を指定した場合は、出力するプリンタの解像度で印刷します。

nDigit

インデックス部分を省いたグループ名に合成 (付加) するインデックス部分の桁数を指定します。0 を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成 (付加) するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetImagePath` , `cwSetImagePathByGrpIdx` , `cwSetImageDIBByBrpIdx` ,
`cwClearItemData` , `cwClearItemDataByGrpIdx` , `cwSetImgStrech` ,
`cwSetImgStrechByGrpIdx` , `cwGetImgStretch` , `cwGetImgStretchByGrpIdx`
`cwSetDIBits` , `cwSetDIBitsByGrpIdx` , `cwSetDPIByGrpIdx`

cwClearItemDataByGrpIdx

アイテムの印刷データを空にします。

テキスト・アイテム：文字列を空にします。

バーコード・アイテム：バーコード及びバーコードデータを空にします

イメージ・アイテム：イメージを空にします。

但し、Fixed(固定項目)が True に設定されているアイテムのデータはクリアされません。

デザイナーで、グループの複製をした時に付けたインデックス(添え字)部分を変数として扱えるようになっています。

C/C++

```
BOOL cwClearItemDataByGrpIdx(  
    HCWREP hCwRep,        //レポートのハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,    //アイテム名  
    INT nDigit,           //インデックスの桁数  
    INT nIdx              //インデックス  
);
```

VB

```
Declare Function cwClearItemDataByGrpIdx Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, ByVal nDigit As Long, ByVal nIdx As Long) As Long
```

Delphi

```
function cwClearItemDataByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; nDigit: Integer; nIdx: Integer): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwClearItemData` , `cwClearAllText` , `cwClearAllBarcode` ,
`cwClearAllImage`

cwSetBoundByGrpIdx

アイテムの位置・大きさを設定します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwSetBoundByGrpIdx(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,   //アイテム名  
    INT *nLeft,          //左位置  
    INT *nTop,           //上位置  
    INT *nWidth,         //幅  
    INT *nHeight,       //高さ  
    INT nDigit,         //インデックスの桁数  
    INT nIndex          //インデックス  
);
```

VB

```
Declare Function cwSetBoundByGrpIdx Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, nLeft As Long, nTop As Long, nWidth As Long, nHeight As  
    Long, ByVal nDigit As Long, ByVal nIndex As Long) As Long
```

Delphi

```
function cwSetBoundByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; nLeft: Pointer; nTop: Pointer; nWidth: Pointer;  
    nHeight: Pointer; nDigit: Integer; nIndex: Integer): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

nLeft

左位置の値を格納している INT 型変数へのポインタ。設定しない場合は、NULL を指定します。

nTop

上位置の値を格納している INT 型変数へのポインタ。設定しない場合は、NULL を指定します。

nWidth

幅の値を格納している INT 型変数へのポインタ。設定しない場合は、NULL を指定

します。

nHeight

高さの値を格納している INT 型変数へのポインタ。設定しない場合は、NULL を指定します。

位置・大きさの値は、レポートの長さの単位により異なります。

単位がミリメートルの場合、1/100 ミリメートルを基本単位とします。インチの場合は、1 / 1000 インチを基本単位とします。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0 を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetBound , cwGetBound , cwGetBoundByGrpIdx , cwSetVisible ,
cwSetVisibleByGrpIdx , cwGetVisible , cwGetVisibleByGrpIdx ,
cwSetForeground , cwSetForegroundByGrpIdx

cwGetBoundByGrpIdx

アイテムの位置・大きさの設定を取得します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwGetBoundByGrpIdx(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,   //アイテム名  
    INT *nLeft,          //左位置  
    INT *nTop,           //上位置  
    INT *nWidth,         //幅  
    INT *nHeight,        //高さ  
    INT nDigit,          //インデックスの桁数  
    INT nIndex           //インデックス  
);
```

VB

```
Declare Function cwGetBoundByGrpIdx Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, nLeft As Long, nTop As Long, nWidth As Long, nHeight As  
    Long, ByVal nDigit As Long, ByVal nIndex As Long) As Long
```

Delphi

```
function cwGetBoundByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; nLeft: Pointer; nTop: Pointer; nWidth: Pointer;  
    nHeight: Pointer; nDigit: Integer; nIndex: Integer): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

nLeft

左位置の値を格納する INT 型変数へのポインタ。取得しない場合は、NULL を指定します。

nTop

上位置の値を格納する INT 型変数へのポインタ。取得しない場合は、NULL を指定します。

nWidth

幅の値を格納する INT 型変数へのポインタ。取得しない場合は、NULL を指定しま

す。

nHeight

高さの値を格納する INT 型変数へのポインタ。取得しない場合は、NULL を指定します。

位置・大きさの値は、レポートの長さの単位により異なります。

単位がミリメートルの場合、1/100 ミリメートルを基本単位とします。インチの場合は、1 / 1000 インチを基本単位とします。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0 を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetBound , cwSetBoundByGrpIdx , cwGetBound , cwSetVisible ,
cwSetVisibleByGrpIdx , cwGetVisible , cwGetVisibleByGrpIdx ,
cwSetForeground , cwSetForegoundByGrpIdx

cwSetPenByGrpIdx

アイテムの線や枠の設定をします。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwSetPenByGrpIdx(  
    HCWREP    hCwRep,        //レポートハンドル  
    LPCSTR    lpGrpName,    //グループ名  
    LPCSTR    lpItemName,   //アイテム名  
    INT       *nPenStyle,   //線の模様  
    COLORREF *crPenColor,  //線の色  
    INT       *nWidth,     //線の太さ  
    INT       *nRound,     //枠の角の丸み  
    INT       *nCapStyle,  //線の終端の形  
    INT       nDigit,      //インデックスの桁数  
    INT       nIdx         //インデックス  
);
```

VB

```
Declare Function cwSetPenByGrpIdx Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal lpGrpName As String, ByVal lpItemName As  
String, nPenStyle As Long, crPenColor As Long, nWidth As Long, nRound  
As Long, nCapStyle As Long, ByVal nDigit As Long, ByVal nIdx As  
Long )As Long
```

Delphi

```
function cwSetPenByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; nPenStyle: Pointer; crPenColor: Pointer;  
nWidth: Pointer; nRound: Pointer; nCapStyle: Pointer; nDigit: Integer;  
nIdx: Integer): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

nPenStyle

線の模様を格納している INT 型へのポインタを指定します。設定しない場合は、NULL を指定します。以下の値のいずれかを指定します。

CW_LS_SOLID	(0) 実線
CW_LS_DASH	(1) 破線
CW_LS_DOT	(2) 点線

CW_LS_DASHDOT (3) 一点鎖線

CW_LS_DASHDOTDOT (4) 二点鎖線

crPenColor

線の色を格納している COLORREF 型へのポインタを指定します。設定しない場合は、NULL を指定します。

nWidth

線の太さの値を格納している INT 型変数へのポインタを指定します。単位は 1/100 ポイント単位です。設定しない場合は、NULL を指定します。

nRound

枠の角の丸みの値を格納している INT 型変数へのポインタを指定します。単位はレポートの長さの単位により異なります。単位がミリメートルの場合、1/100 ミリメートルを基本単位とします。インチの場合は、1 / 1000 インチを基本単位とします。設定しない場合は、NULL を指定します。

nCapStyle (Windows2000/XP のみ有効)

線の終端の形の値を格納している INT 型変数へのポインタを指定します。設定しない場合は、NULL を指定します。

CW_ENDCAP_ROUND (0) 終端を丸くします。

CW_ENDCAP_SQUARE (1) 終端を四角にします。

CW_ENDCAP_FLAT (2) 終端を平らにします。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0 を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetPen , cwGetPen, cwGetPenByGrpIdx , cwSetBrush ,
cwSetBrushByGrpIdx , cwGetBrush , cwGetBrushByGrpIdx

cwGetPenByGrpIdx

アイテムの線や枠の設定を取得します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwGetPenByGrpIdx(  
    HCWREP hCwRep,          //レポートハンドル  
    LPCSTR lpGrpName,      //グループ名  
    LPCSTR lpItemName,    //アイテム名  
    INT *nPenStyle,       //線の模様  
    COLORREF *crPenColor, //線の色  
    INT *nWidth,          //線の太さ  
    INT *nRound,          //枠の角の丸み  
    INT *nCapStyle,       //線の終端の形  
    INT nDigit,           //インデックスの桁数  
    INT nIndex            //インデックス  
);
```

VB

```
Declare Function cwGetPenByGrpIdx Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal lpGrpName As String, ByVal lpItemName As  
String, nPenStyle As Long, crPenColor As Long, nWidth As Long, nRound  
As Long, nCapStyle As Long, ByVal nDigit As Long, ByVal nIndex As  
Long) As Long
```

Delphi

```
function cwGetPenByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; nPenStyle: Pointer; crPenColor: Pointer;  
nWidth: Pointer; nRound: Pointer; nCapStyle: Pointer; nDigit: Integer;  
nIndex: Integer): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

nPenStyle

線の模様を格納する INT 型へのポインタを指定します。取得しない場合は、NULL を指定します。以下のいずれかの値が取得できます。

CW_LS_SOLID	(0)	実線
CW_LS_DASH	(1)	破線
CW_LS_DOT	(2)	点線

CW_LS_DASHDOT (3) 一点鎖線

CW_LS_DASHDOTDOT (4) 二点鎖線

crPenColor

線の色を格納する COLORREF 型へのポインタを指定します。取得しない場合は、NULL を指定します。

nWidth

線の太さの値を格納する INT 型変数へのポインタを指定します。単位は 1/100 ポイント単位です。設取得しない場合は、NULL を指定します。

nRound

枠の角の丸みの値を格納する INT 型変数へのポインタを指定します。単位はレポートの長さの単位により異なります。単位がミリメートルの場合、1/100 ミリメートルを基本単位とします。インチの場合は、1 / 1000 インチを基本単位とします。取得しない場合は、NULL を指定します。

nCapStyle (Windows2000/XP のみ有効)

線の終端の形の値を格納する INT 型変数へのポインタを指定します。取得しない場合は、NULL を指定します。以下のいずれかの値が取得できます。

CW_ENDCAP_ROUND (0) 終端を丸くします。

CW_ENDCAP_SQUARE (1) 終端を四角にします。

CW_ENDCAP_FLAT (2) 終端を平らにします。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0 を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetPen , cwSetPenByGrpIdx , cwGetPen , cwSetBrush ,
cwSetBrushByGrpIdx , cwGetBrush , cwGetBrushByGrpIdx

cwSetBrushByGrpIdx

アイテムの背景の塗りつぶし模様や色の設定をします。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwSetBrushByGrpIdx(  
    HCWREP    hCwRep,        //レポートハンドル  
    LPCSTR    lpGrpName,    //グループ名  
    LPCSTR    lpItemName,   //アイテム名  
    INT       *nBrushStyle, //塗りつぶしの模様  
    COLORREF  *crBrushColor, //塗りつぶしの色  
    INT       nDigit,       //インデックスの桁数  
    INT       nIdx          //インデックス  
);
```

VB

```
Declare Function cwSetBrushByGrpIdx Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, nBrushStyle As Long, crBrushColor As Long, ByVal nDigit As  
    Long, ByVal nIdx As Long) As Long
```

Delphi

```
function cwSetBrushByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; nBrushStyle: Pointer; crBrushColor: Pointer;  
    nDigit: Integer; nIdx: Integer): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

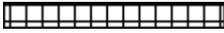
インデックス部分を省いたグループ名へのポインタを指定します。

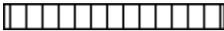
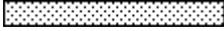
lpItemName

アイテム名へのポインタを指定します。

nBrushStyle

塗りつぶしの模様を格納している INT 型へのポインタを指定します。設定しない場合は、NULL を指定します。以下の値のいずれかを指定します。

CW_BS_NONE	(0)	塗りつぶさない。
CW_BS_SOLID	(1)	塗りつぶす。
CW_BS_PATTERN1	(2)	
CW_BS_PATTERN2	(3)	
CW_BS_PATTERN3	(4)	
CW_BS_PATTERN4	(5)	

CW_BS_PATTERN5	(6)	
CW_BS_PATTERN6	(7)	
CW_BS_PATTERN7	(8)	
CW_BS_PATTERN8	(9)	
CW_BS_PATTERN9	(10)	
CW_BS_PATTERN10	(11)	

crBrushColor

塗りつぶしの色を格納している COLORREF 型へのポインタを指定します。設定しない場合は、NULL を指定します。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0 を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetPen , cwSetPenByGrpIdx , cwGetPen , cwGetPenByGrpIdx ,
cwSetBrush , cwGetBrush , cwGetBrushByGrpIdx

cwGetBrushByGrpIdx

アイテムの背景の塗りつぶし模様や色の設定を取得します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwGetBrushByGrpIdx(  
    HCWREP    hCwRep,          //レポートハンドル  
    LPCSTR    lpGrpName,      //グループ名  
    LPCSTR    lpItemName,     //アイテム名  
    INT       *nBrushStyle,   //塗りつぶしの模様  
    COLORREF  *crBrushColor,  //塗りつぶしの色  
    INT       nDigit,         //インデックスの桁数  
    INT       nIdx            //インデックス  
);
```

VB

```
Declare Function cwGetBrushByGrpIdx Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, nBrushStyle As Long, crBrushColor As Long, ByVal nDigit As  
    Long, ByVal nIdx As Long) As Long
```

Delphi

```
function cwGetBrushByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; nBrushStyle: Pointer; crBrushColor: Pointer;  
    nDigit: Integer; nIdx: Integer): Integer; stdcall;  
external 'cwRepEng.dll'
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

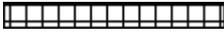
インデックス部分を省いたグループ名へのポインタを指定します。

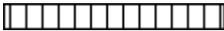
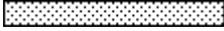
lpItemName

アイテム名へのポインタを指定します。

nBrushStyle

塗りつぶしの模様を格納する INT 型へのポインタを指定します。取得しない場合は、NULL を指定します。以下の値のいずれかが取得できます。

CW_BS_NONE	(0)	塗りつぶさない。
CW_BS_SOLID	(1)	塗りつぶす。
CW_BS_PATTERN1	(2)	
CW_BS_PATTERN2	(3)	
CW_BS_PATTERN3	(4)	
CW_BS_PATTERN4	(5)	

CW_BS_PATTERN5	(6)	
CW_BS_PATTERN6	(7)	
CW_BS_PATTERN7	(8)	
CW_BS_PATTERN8	(9)	
CW_BS_PATTERN9	(10)	
CW_BS_PATTERN10	(11)	

crBrushColor

塗りつぶしの色を格納する COLORREF 型へのポインタを指定します。取得しない場合は、NULL を指定します。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0 を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetPen , cwSetPenByGrpIdx , cwGetPen , cwGetPenByGrpIdx ,
cwSetBrush ,cwSetBrushByGrpIdx , cwGetBrush

cwSetFontByGrpIdx

アイテムのフォント情報を設定をします。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwSetFontByGrpIdx(  
    HCWREP    hCwRep,           //レポートハンドル  
    LPCSTR    lpGrpName,       //グループ名  
    LPCSTR    lpItemName,      //アイテム名  
    LPCSTR    lpFontName,      //フォント名  
    INT       *nFontStyle,     //フォントスタイル  
    INT       *nFontSize,      //フォントサイズ  
    COLORREF  *crFontColor,    //フォントの色  
    INT       nDigit,          //インデックスの桁数  
    INT       nIdx             //インデックス  
);
```

VB

```
Declare Function cwSetFontByGrpIdx Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal lpGrpName As String, ByVal lpItemName As String, ByVal  
lpFontName As String, nFontStyle As Long, nFontSize As  
Long, crFontColor As Long, ByVal nDigit As Long, ByVal nIdx As Long) As  
Long
```

Delphi

```
function cwSetFontByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; lpFontName: PChar; nFontStyle: Pointer;  
nFontSize: Pointer; crFontColor: Pointer; nDigit: Integer;  
nIdx: Integer): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

lpFontName

フォント名へのポインタを指定します。設定しない場合は、NULL を指定します。

nFontStyle

フォントスタイルの値を格納している INT 型へのポインタを指定します。設定しない場合は、NULL を指定します。以下の値の組み合わせで設定します。

CW_FS_BOLD (1) 太字

CW_FS_ITALIC (2) 斜体

CW_FS_UNDERLINE (4) 下線

nFontSize

フォントサイズを格納している INT 型へのポインタを指定します。設定しない場合は、NULL を指定します。フォントサイズの単位は 1/10 ポイント単位で指定します。

crFontColor

フォント色を格納している COLORREF 型へのポインタを指定します。設定しない場合は、NULL を指定します。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0 を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetFont , cwGetFont , cwGetFontByGrpIdx , cwSetTextAlign ,
cwSetTextAlignByGrpIdx , cwGetTextAlign , cwGetTextAlignByGrpIdx ,
cwSetSpace , cwSetSpaceByGrpIdx , cwGetSpace , cwGetSpaceByGrpIdx ,
cwSetTextEscapement , cwSetTextEscapementByGrpIdx , cwGetTextEscapement ,
cwGetTextEscapementByGrpIdx , cwGetTextMetrics , cwGetTextMetricsByGrpIdx ,
cwSetWordwrap , cwSetWordwrapByGrpIdx , cwGetWordwrap , cwGetWordwrapByGrpIdx

cwGetFontByGrpIdx

アイテムのフォント情報を取得します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwGetFontByGrpIdx(  
    HCWREP h CwRep,           //レポートハンドル  
    LPCSTR lpGrpName,        //グループ名  
    LPCSTR lpItemName,       //アイテム名  
    LPCSTR lpFontName,       //フォント名  
    INT *nFontStyle,         //フォントスタイル  
    INT *nFontSize,          //フォントサイズ  
    COLORREF *crFontColor,   //フォントの色  
    INT nDigit,              //インデックスの桁数  
    INT nIdx                  //インデックス  
);
```

VB

```
Declare Function cwGetFontByGrpIdx Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal lpGrpName As String, ByVal lpItemName As String, ByVal  
lpFontName As String, nFontStyle As Long, nFontSize As  
Long, crFontColor As Long, ByVal nDigit As Long, ByVal nIdx As Long) As  
Long
```

Delphi

```
function cwGetFontByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; lpFontName: PChar; nFontStyle: Pointer;  
nFontSize: Pointer; crFontColor: Pointer; nDigit: Integer;  
nIdx: Integer): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

lpFontName

フォント名を格納するバッファのポインタを指定します。必要なバッファの大きさは65バイトです。取得しない場合は、NULLを指定します。

nFontStyle

フォントスタイルの値を格納するINT型へのポインタを指定します。取得しない場合は、NULLを指定します。以下の値の組み合わせの値が取得できます。

CW_FS_BOLD (1) 太字

CW_FS_ITALIC (2) 斜体

CW_FS_UNDERLINE (4) 下線

nFontSize

フォントサイズを格納する INT 型へのポインタを指定します。取得しない場合は、NULL を指定します。フォントサイズの単位は 1/10 ポイント単位です。

crFontColor

フォント色を格納する COLORREF 型へのポインタを指定します。取得しない場合は、NULL を指定します。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0 を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetFont , cwSetFontByGrpIdx , cwGetFont , cwSetTextAlign ,
cwSetTextAlignByGrpIdx , cwGetTextAlign , cwGetTextAlignByGrpIdx ,
cwSetSpace , cwSetSpaceByGrpIdx , cwGetSpace , cwGetSpaceByGrpIdx ,
cwSetTextEscapement , cwSetTextEscapementByGrpIdx , cwGetTextEscapement ,
cwGetTextEscapementByGrpIdx , cwGetTextMetrics , cwGetTextMetricsByGrpIdx ,
cwSetWordwrap , cwSetWordwrapByGrpIdx , cwGetWordwrap , cwGetWordwrapByGrpIdx

cwSetTextAlignByGrpIdx

テキスト・アイテムの文字の寄せを設定します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwSetTextAlignByGrpIdx(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,    //アイテム名  
    INT *nHorzAlign,      //文字の寄せ  
    INT *nVertAlign,      //行の寄せ  
    INT nDigit,           //インデックスの桁数  
    INT nIdx               //インデックス  
);
```

VB

```
Declare Function cwSetTextAlignByGrpIdx Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, nHorzAlign As Long, nVertAlign As Long, ByVal nDigit As  
    Long, ByVal nIdx As Long) As Long
```

Delphi

```
function cwSetTextAlignByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; nHorzAlign: Pointer; nVertAlign: Pointer; nDigit:  
    Integer; nIdx: Integer): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

nHorzAlign

文字の寄せ設定を格納している INT 型へのポインタを指定します。設定しない場合は、NULL を指定します。以下のいずれかの値を指定します。

CW_ALR_LEFT (0) 左寄せ（上寄せ）

CW_ALR_CENTER (1) 中央寄せ

CW_ALR_RIGHT (2) 右寄せ（下寄せ）

CW_ALR_EQUAL (3) 均等配置

括弧内は縦書きの場合。

nVertAlign

行の寄せ設定を格納している INT 型へのポインタを指定します。設定しない場合

は、NULL を指定します。以下のいずれかの値を指定します。

CW_ATB_TOP (0) 上寄せ (右寄せ)
CW_ATB_CENTER (1) 中央寄せ
CW_ATB_BOTTOM (2) 下寄せ (左寄せ)
CW_ATB_EQUAL (3) 均等配置

括弧内は縦書きの場合。

nDigit

インデックス部分を省いたグループ名に合成 (付加) するインデックス部分の桁数を指定します。0 を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成 (付加) するインデックスの値を指定します。

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetFont , cwSetFontByGrpIdx , cwGetFont , cwGetFontByGrpIdx ,
cwSetTextAlign , cwGetTextAlign , cwGetTextAlignByGrpIdx ,
cwSetSpace , cwSetSpaceByGrpIdx , cwGetSpace , cwGetSpaceByGrpIdx
cwSetTextEscapement , cwSetTextEscapementByGrpIdx , cwGetTextEscapement ,
cwGetTextEscapementByGrpIdx , cwGetTextMetrics , cwGetTextMetricsByGrpIdx ,
cwSetWordwrap , cwSetWordwrapByGrpIdx , cwGetWordwrap , cwGetWordwrapByGrpIdx

cwGetTextAlignByGrpIdx

テキスト・アイテムの文字の寄せを設定します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwGetTextAlignByGrpIdx(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,    //アイテム名  
    INT *nHorzAlign,      //文字の寄せ  
    INT *nVertAlign,      //行の寄せ  
    INT nDigit,           //インデックスの桁数  
    INT nIdx              //インデックス  
);
```

VB

```
Declare Function cwGetTextAlignByGrpIdx Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, nHorzAlign As Long, nVertAlign As Long, ByVal nDigit As  
    Long, ByVal nIdx As Long) As Long
```

Delphi

```
function cwGetTextAlignByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; nHorzAlign: Pointer; nVertAlign: Pointer; nDigit:  
    Integer; nIdx: Integer): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

nHorzAlign

文字の寄せ設定を格納する INT 型へのポインタを指定します。取得しない場合は、NULL を指定します。

CW_ALR_LEFT (0) 左寄せ（上寄せ）

CW_ALR_CENTER (1) 中央寄せ

CW_ALR_RIGHT (2) 右寄せ（下寄せ）

CW_ALR_EQUAL (3) 均等配置

括弧内は縦書きの場合。

nVertAlign

行の寄せ設定を格納する INT 型へのポインタを指定します。取得しない場合は、NULL を指定します。

CW_ATB_TOP (0) 上寄せ (右寄せ)
CW_ATB_CENTER (1) 中央寄せ
CW_ATB_BOTTOM (2) 下寄せ (左寄せ)
CW_ATB_EQUAL (3) 均等配置

括弧内は縦書きの場合。

nDigit

インデックス部分を省いたグループ名に合成 (付加) するインデックス部分の桁数を指定します。0 を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成 (付加) するインデックスの値を指定します。

戻り値

関数が成功すると、0 以外の数字が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetFont , cwSetFontByGrpIdx , cwGetFont , cwGetFontByGrpIdx ,
cwSetTextAlign , cwSetTextAlignByGrpIdx , cwGetTextAlign ,
cwSetSpace , cwSetSpaceByGrpIdx , cwGetSpace , cwGetSpaceByGrpIdx ,
cwSetTextEscapement , cwSetTextEscapementByGrpIdx , cwGetTextEscapement ,
cwGetTextEscapementByGrpIdx , cwGetTextMetrics , cwGetTextMetricsByGrpIdx ,
cwSetWordwrap , cwSetWordwrapByGrpIdx , cwGetWordwrap , cwGetWordwrapByGrpIdx

cwSetSpaceByGrpIdx

テキスト・アイテムの文字・行の間隔を設定します。デザイナーで、グループの複製をした時に付けたインデックス(添え字)部分を変数として扱えるようになっています。

C/C++

```
BOOL cwSetSpaceByGrpIdx(  
    HCWREP hCwRep,          //レポートハンドル  
    LPCSTR lpGrpName,      //グループ名  
    LPCSTR lpItemName,     //アイテム名  
    INT *nCharSpace,       //文字の寄せ  
    INT *nLineSpace,       //行の寄せ  
    INT nDigit,            //インデックスの桁数  
    INT nIndex             //インデックス  
);
```

VB

```
Declare Function cwSetSpaceByGrpIdx Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, nCharSpace As Long, nLineSpace As Long, ByVal nDigit As  
    Long, ByVal nIndex As Long) As Long
```

Delphi

```
function cwSetSpaceByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; nCharSpace: Pointer; nLineSpace: Pointer; nDigit:  
    Integer; nIndex: Integer): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

nCharSpace

文字の間隔を格納している INT 型へのポインタを指定します。設定しない場合は、NULL を指定します。設定する値の単位はレポートの長さの単位により異なります。単位がミリメートルの場合、1/100 ミリメートルを基本単位とします。インチの場合は、1 / 1000 インチを基本単位とします。

nLineSpace

行の間隔を格納している INT 型へのポインタを指定します。設定しない場合は、NULL を指定します。設定する値の単位はレポートの長さの単位により異なります。単位がミリメートルの場合、1/100 ミリメートルを基本単位とします。インチの場合は、1 / 1000 インチを基本単位とします。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetFont , cwSetFontByGrpIdx , cwGetFont , cwGetFontByGrpIdx ,
cwSetTextAlign , cwSetTextAlignByGrpIdx , cwGetTextAlign ,
cwGetTextAlignByGrpIdx , cwSetSpace , cwGetSpace
cwGetSpaceByGrpIdx , cwSetTextEscapement , cwSetTextEscapementByGrpIdx ,
cwGetTextEscapement , cwGetTextEscapementByGrpIdx , cwGetTextMetrics
cwGetTextMetricsByGrpIdx , cwSetWordwrap , cwSetWordwrapByGrpIdx ,
cwGetWordwrap , cwGetWordwrapByGrpIdx

cwGetSpaceByGrpIdx

テキスト・アイテムの文字・行の間隔の設定を取得します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwGetSpaceByGrpIdx(  
    HCWREP hCwRep,          //レポートハンドル  
    LPCSTR lpGrpName,      //グループ名  
    LPCSTR lpItemName,     //アイテム名  
    INT *nCharSpace,       //文字の寄せ  
    INT *nLineSpace,       //行の寄せ  
    INT nDigit,            //インデックスの桁数  
    INT nIndex             //インデックス  
);
```

VB

```
Declare Function cwGetSpaceByGrpIdx Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, nCharSpace As Long, nLineSpace As Long, ByVal nDigit As  
    Long, ByVal nIndex As Long) As Long
```

Delphi

```
function cwGetSpaceByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; nCharSpace: Pointer; nLineSpace: Pointer; nDigit:  
    Integer; nIndex: Integer): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

nCharSpace

文字の間隔を格納する INT 型へのポインタを指定します。取得しない場合は、NULL を指定します。取得する値の単位はレポートの長さの単位により異なります。単位がミリメートルの場合、1/100 ミリメートルを基本単位とします。インチの場合は、1 / 1000 インチを基本単位とします。

nLineSpace

行の間隔を格納する INT 型へのポインタ指定します。取得しない場合は、NULL を指定します。取得する値の単位はレポートの長さの単位により異なります。単位がミリメートルの場合、1/100 ミリメートルを基本単位とします。インチの場合は、1 / 1000 インチを基本単位とします。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetFont , cwSetFontByGrpIdx , cwGetFont , cwGetFontByGrpIdx ,
cwSetTextAlign , cwSetTextAlignByGrpIdx , cwGetTextAlign ,
cwGetTextAlignByGrpIdx , cwSetSpace , cwSetSpaceByGrpIdx ,
cwGetSpace , cwSetTextEscapement , cwSetTextEscapementByGrpIdx ,
cwGetTextEscapement , cwGetTextEscapementByGrpIdx , cwGetTextMetrics
cwGetTextMetricsByGrpIdx , cwSetWordwrap , cwSetWordwrapByGrpIdx ,
cwGetWordwrap , cwGetWordwrapByGrpIdx

cwSetTextEscapementByGrpIdx

テキスト・アイテムの文字の横書き・縦書きを設定します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwSetTextEscapementByGrpIdx(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,    //アイテム名  
    INT nEscMode,         //横書き・縦書きの設定  
    INT nDigit,           //インデックスの桁数  
    INT nIndex            //インデックス  
);
```

VB

```
Declare Function cwSetTextEscapementByGrpIdx Lib  
"cwRepEng.dll" (ByVal hCwRep As Long, ByVal lpGrpName As  
String, ByVal lpItemName As String, ByVal nEscMode As Long, ByVal  
nDigit As Long, ByVal nIndex As Long) As Long
```

Delphi

```
function cwSetTextEscapementByGrpIdx(hCwRep: Integer;  
lpGrpName: PChar; lpItemName: PChar; nEscMode: Integer; nDigit:  
Integer; nIndex: Integer): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

nEscMode

横書き・縦書きの値を指定します。以下のいずれかの値を指定します。

0:横書き 1:縦書き

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetFont` , `cwSetFontByGrpIdx` , `cwGetFont` , `cwGetFontByGrpIdx` ,
`cwSetTextAlign` , `cwSetTextAlignByGrpIdx` , `cwGetTextAlign` ,
`cwGetTextAlignByGrpIdx` , `cwSetSpace` , `cwSetSpaceByGrpIdx` ,
`cwGetSpace` , `cwGetSpaceByGrpIdx` , `cwSetTextEscapement`
`cwGetTextEscapement` , `cwGetTextEscapementByGrpIdx` , `cwGetTextMetrics`
`cwGetTextMetricsByGrpIdx` , `cwSetWordwrap` , `cwSetWordwrapByGrpIdx` ,
`cwGetWordwrap` , `cwGetWordwrapByGrpIdx`

cwGetTextEscapementByGrpIdx

テキスト・アイテムの文字の横書き・縦書きの設定を取得します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwGetTextEscapementByGrpIdx(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,    //グループ名  
    LPCSTR lpItemName,   //アイテム名  
    INT *nEscMode,       //横書き・縦書きの設定  
    INT nDigit,          //インデックスの桁数  
    INT nIdx             //インデックス  
);
```

VB

```
Declare Function cwSetTextEscapementByGrpIdx Lib "cwRepEng.dll"  
(ByVal hCwRep As Long,ByVal lpGrpName As String,ByVal  
lpItemName As String,ByVal nEscMode As Long,ByVal nDigit As  
Long,ByVal nIdx As Long)As Long
```

Delphi

```
function cwSetTextEscapementByGrpIdx(hCwRep: Integer;  
lpGrpName: PChar; lpItemName: PChar; nEscMode: Integer; nDigit:  
Integer; nIdx: Integer ): Integer;stdcall;external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

nEscMode

横書き・縦書きの値を指定します。以下のいずれかの値を指定します。

0:横書き 1:縦書き

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetFont` , `cwSetFontByGrpIdx` , `cwGetFont` , `cwGetFontByGrpIdx` ,
`cwSetTextAlign` , `cwSetTextAlignByGrpIdx` , `cwGetTextAlign` ,
`cwGetTextAlignByGrpIdx` , `cwSetSpace` , `cwSetSpaceByGrpIdx` ,
`cwGetSpace` , `cwGetSpaceByGrpIdx` , `cwSetTextEscapement`
`cwGetTextEscapement` , `cwGetTextEscapementByGrpIdx` , `cwGetTextMetrics`
`cwGetTextMetricsByGrpIdx` , `cwSetWordwrap` , `cwSetWordwrapByGrpIdx` ,
`cwGetWordwrap` , `cwGetWordwrapByGrpIdx`

cwGetTextMetricsByGrpIdx

テキストアイテムの印刷する文字列の高さ・幅を取得します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwGetTextMetricsByGrpIdx(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,    //グループ名  
    LPCSTR lpItemName,   //アイテム名  
    DOUBLE *dbWidth,     //文字列の高さ  
    DOUBLE *dbHeight    //文字列の幅  
    INT nDigit,         //インデックスの桁数  
    INT nIndex         //インデックス  
);
```

VB

```
Public Declare Function cwGetTextMetricsByGrpIdxLib "cwRepEng.  
dll"(ByVal hCwRep As Long,ByVal lpGrpName As String,ByVal  
lpItemName As String,dbWidth As Double,dbHeight As Double,ByVal  
nDigit As Long,ByVal nIndex As Long )As Long
```

Delphi

```
function cwGetTextMetricsByGrpIdx(hCwRep: Integer;  
lpGrpName: PChar; lpItemName: PChar; dbWidth: Pointer;  
dbHeight: Pointer; nDigit: Integer; nIndex: Integer ): Integer;  
stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

dbWidth

文字列の幅を取得する DOUBLE 型のポインタを指定します。長さの単位は、レポートの長さの単位に従い、ミリメートル/インチとなります。文字列の幅は、指定した文字間隔が含まれたものが返ります。

dbHeight

文字列の高さを取得する DOUBLE 型のポインタを指定します。長さの単位は、レポートの長さの単位に従い、ミリメートル/インチとなります。指定した行の間隔は含まれません。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetFont` , `cwSetFontByGrpIdx` , `cwGetFont` , `cwGetFontByGrpIdx` ,
`cwSetTextAlign` , `cwSetTextAlignByGrpIdx` , `cwGetTextAlign` ,
`cwGetTextAlignByGrpIdx` , `cwSetSpace` , `cwSetSpaceByGrpIdx` ,
`cwGetSpace` , `cwGetSpaceByGrpIdx` , `cwSetTextEscapement`
`cwGetTextEscapement` , `cwGetTextEscapementByGrpIdx` , `cwGetTextMetrics`
`cwSetWordwrap` , `cwSetWordwrapByGrpIdx` , `cwGetWordwrap` , `cwGetWordwrapByGrpIdx`

cwSetWordwrapByGrpIdx

テキスト・アイテムのテキストの折り返しを設定します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwSetWordwrapByGrpIdx(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,   //アイテム名  
    BOOL bWordwrap,      //折り返しの設定  
    INT nDigit,          //インデックスの桁数  
    INT nIdx             //インデックス  
);
```

VB

```
Declare Function cwSetWordwrapByGrpIdx Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, ByVal bWordwrap As Long, ByVal nDigit As Long, ByVal nIdx  
    As Long) As Long
```

Delphi

```
function cwSetWordwrapByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; bWordwrap: Integer; nDigit: Integer; nIdx: Integer  
    ): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

bWordwrap

テキストの折り返しの設定を指定します。以下のいずれかの値を指定します。

0 : : テキストを折り返さない

0 以外 : テキストを折り返す

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetFont` , `cwSetFontByGrpIdx` , `cwGetFont` , `cwGetFontByGrpIdx` ,
`cwSetTextAlign` , `cwSetTextAlignByGrpIdx` , `cwGetTextAlign` ,
`cwGetTextAlignByGrpIdx` , `cwSetSpace` , `cwSetSpaceByGrpIdx` ,
`cwGetSpace` , `cwGetSpaceByGrpIdx` , `cwSetTextEscapement`
`cwGetTextEscapement` , `cwGetTextEscapementByGrpIdx` , `cwGetTextMetrics`
`cwGetTextMetricsByGrpIdx` , `cwSetWordwrap` , `cwGetWordwrap` ,
`cwGetWordwrapByGrpIdx`

cwGetWordwrapByGrpIdx

テキスト・アイテムのテキストの折り返しの設定を取得します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwGetWordwrapByGrpIdx(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,    //アイテム名  
    BOOL *bWordwrap,     //折り返しの設定  
    INT nDigit,          //インデックスの桁数  
    INT nIdx             //インデックス  
);
```

VB

```
Declare Function cwGetWordwrapByGrpIdx Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, bWordwrap As Long, ByVal nDigit As Long, ByVal nIdx As Lon  
g) As Long
```

Delphi

```
function cwGetWordwrapByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; bWordwrap: Pointer; nDigit: Integer; nIdx: Integer  
): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

bWordwrap

テキストの折り返しの設定を取得する BOOL 型のポインタを指定します。以下のいずれかの値を取得できます。

0 : テキストを折り返さない

0 以外 : テキストを折り返す

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetFont` , `cwSetFontByGrpIdx` , `cwGetFont` , `cwGetFontByGrpIdx` ,
`cwSetTextAlign` , `cwSetTextAlignByGrpIdx` , `cwGetTextAlign` ,
`cwGetTextAlignByGrpIdx` , `cwSetSpace` , `cwSetSpaceByGrpIdx` ,
`cwGetSpace` , `cwGetSpaceByGrpIdx` , `cwSetTextEscapement`
`cwGetTextEscapement` , `cwGetTextEscapementByGrpIdx` , `cwGetTextMetrics` ,
`cwGetTextMetricsByGrpIdx` , `cwSetWordwrap` , `cwSetWordwrapByGrpIdx` ,
`cwGetWordwrap`

cwSetBarcodePropByGrpIdx

バーコード・アイテムの詳細な設定をします。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL WINAPI cwSetBarcodePropByGrpIdx(  
    HCWREP        hCwRep,          //レポートハンドル  
    LPCSTR        lpGrpName,       //グループ名  
    LPCSTR        lpItemName,     //アイテム名  
    CW_BARCODEDEF *lpBarcodeDef,  //バーコードプロパティ  
    INT           nDigit,         //インデックスの桁数  
    INT           nIndex          //インデックス  
);
```

VB

```
Declare Function cwSetBarcodePropByGrpIdx Lib "cwRepEng.dll"  
(ByVal hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName  
As String, lpBarcodeDef As CW_BARCODEDEF, ByVal nDigit As Long, ByVal  
nIndex As Long) As Long
```

Delphi

```
function cwSetBarcodePropByGrpIdx(hCwRep: Integer; lpGrpName:  
PChar; lpItemName: PChar; lpBarcodeDef: Pointer; nDigit: Integer;  
nIndex: Integer): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

lpBarcodeDef

バーコードプロパティ・データブロックへのポインタを指定します。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIndex

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetText , cwSetTextByGrpIdx , cwGetText , cwGetTextByGrpIdx ,
cwSetBarCodeProp , cwGetBarCodeProp , cwGetBarCodePropByGrpIdx

cwGetBarcodePropByGrpIdx

バーコード・アイテムの詳細な設定を取得します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL WINAPI cwGetBarcodePropByGrpIdx(  
    HCWREP        hCwRep,          //レポートハンドル  
    LPCSTR        lpGrpName,       //グループ名  
    LPCSTR        lpItemName,      //アイテム名  
    CW_BARCODEDEF *lpBarcodeDef,   //バーコードプロパティ  
    INT nDigit,          //インデックスの桁数  
    INT nIdx            //インデックス  
);
```

VB

```
Declare Function cwGetBarcodePropByGrpIdx Lib "cwRepEng.dll"  
(ByVal hCwRep As Long,ByVal lpGrpName As String,ByVal lpItemName  
As String,lpBarcodeDef As CW_BARCODEDEF,ByVal nDigit As Long,ByVal  
nIdx As Long )As Long
```

Delphi

```
function cwGetBarcodePropByGrpIdx(hCwRep: Integer; lpGrpName:  
PChar; lpItemName: PChar; lpBarcodeDef: Pointer; nDigit: Integer;  
nIdx: Integer ): Integer;stdcall;external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

lpBarcodeDef

取得するバーコードプロパティ・データブロックを格納するバッファのポインタを指定します。

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetText , cwSetTextByGrpIdx , cwGetText , cwGetTextByGrpIdx ,
cwSetBarCodeProp , cwSetBarCodePropByGrpIdx , cwGetBarCodeProp

cwSetImgStretchByGrpIdx

イメージ・アイテムの領域に収まるようにイメージを拡大・縮小するかを設定をします。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwSetImgStretchByGrpIdx(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,    //アイテム名  
    BOOL bImgStretch,    //ストレッチ  
    INT nDigit,          //インデックスの桁数  
    INT nIdx             //インデックス  
);
```

VB

```
Declare Function cwSetImgStretchByGrpIdx Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, ByVal bImgStretch As Long, ByVal nDigit As Long, ByVal nIdx  
    As Long) As Long
```

Delphi

```
function cwSetImgStretchByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; bImgStretch: Integer; nDigit: Integer; nIdx:  
    Integer): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

bStretch

イメージアイテムの領域に納まるようにイメージを拡大縮小するかを指定します。

0 : 変形しない

0以外 : 収まるようにする

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetImagePath` , `cwSetImagePathByGrpIdx` , `cwSetImageDIB` ,
`cwSetImageDIBByBrpIdx` , `cwSetImgStrech` , `cwGetImgStretch`
`cwGetImgStretchByGrpIdx`

cwGetImgStretchByGrpIdx

イメージ・アイテムの領域に収まるようにイメージを拡大・縮小するかの設定を取得します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwGetImgStretchByGrpIdx(  
    HCWREP hCwRep,          //レポートハンドル  
    LPCSTR lpGrpName,      //グループ名  
    LPCSTR lpItemName,     //アイテム名  
    BOOL *bImgStretch,    //ストレッチ  
    INT nDigit,           //インデックスの桁数  
    INT nIndex            //インデックス  
);
```

VB

```
Declare Function cwGetImgStretchByGrpIdx Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, bImgStretch As Long, ByVal nDigit As Long, ByVal nIndex As  
    Long) As Long
```

Delphi

```
function cwGetImgStretchByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; bImgStretch: Pointer; nDigit: Integer; nIndex:  
    Integer): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

bStretch

イメージアイテムの領域に納まるようにイメージを拡大縮小するかの設定を取得する BOOL 型のポインタを指定します。以下のいずれかの値が取得できます。

0 : 変形しない

0 以外 : 収まるようにする

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetImagePath` , `cwSetImagePathByGrpIdx` , `cwSetImageDIB` ,
`cwSetImageDIBByBrpIdx` , `cwSetImgStrech` , `cwSetImgStrechByGrpIdx`
`cwGetImgStretch`

cwSetVisibleByGrpIdx

アイテムを表示するか否かを設定をします。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwSetVisibleByGrpIdx(  
HCWREP hCwRep,      //レポートハンドル  
LPCSTR lpGrpName,   //グループ名  
LPCSTR lpItemName,  //アイテム名  
BOOL bVisible,      //表示・非表示  
INT nDigit,         //インデックスの桁数  
INT nIdx            //インデックス  
);
```

VB

```
Declare Function cwSetVisibleByGrpIdx Lib "cwRepEng.dll" (ByVal  
hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
String, ByVal bVisible As Long, ByVal nDigit As Long, ByVal nIdx As  
Long) As Long
```

Delphi

```
function cwSetVisibleByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; bVisible: Integer; nDigit: Integer; nIdx: Integer):  
Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

bVisible

アイテムを表示・非表示するかを指定します。以下のいずれかの値を指定します。

0 : 表示しない

0以外 : 表示する

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetImagePath` , `cwSetImagePathByGrpIdx` , `cwSetImageDIB` ,
`cwSetImageDIBByBrpIdx` , `cwSetImgStretch` , `cwSetImgStretchByGrpIdx`
`cwGetImgStretch`

cwGetVisibleByGrpIdx

アイテムを表示するか否かの設定を取得します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwSetVisibleByGrpIdx(  
HCWREP hCwRep,      //レポートハンドル  
LPCSTR lpGrpName,   //グループ名  
LPCSTR lpItemName,  //アイテム名  
BOOL bVisible,      //表示・非表示  
INT nDigit,         //インデックスの桁数  
INT nIdx            //インデックス  
);
```

VB

```
Declare Function cwSetVisibleByGrpIdx Lib "cwRepEng.dll" (ByVal  
hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
String, ByVal bVisible As Long, ByVal nDigit As Long, ByVal nIdx As  
Long) As Long
```

Delphi

```
function cwSetVisibleByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; bVisible: Integer; nDigit: Integer; nIdx: Integer):  
Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

bVisible

アイテムを表示・非表示するかの設定を格納する BOOL 型のポインタを指定します。

0 : 表示しない

0 以外 : 表示する

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwSetBound , cwSetBoundByGrpIdx , cwGetBound ,cwGetBoundByGrpIdx
cwSetVisible , cwSetVisibleByGrpIdx , cwGetVisible
cwSetForeground , cwSetForegoundByGrpIdx

cwSetForegroundByGrpIdx

アイテムを前面・背面に移動させます。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwSetForegroundByGrpIdx(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,    //アイテム名  
    BOOL bFore,          //前面・背面移動  
    INT nDigit,          //インデックスの桁数  
    INT nIdx              //インデックス  
);
```

VB

```
Declare Function cwSetForegroundByGrpIdx Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
    String, ByVal bFore As Long, ByVal nDigit As Long, ByVal nIdx As  
    Long) As Long
```

Delphi

```
function cwSetForegroundByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
    lpItemName: PChar; bFore: Integer; nDigit: Integer; nIdx: Integer):  
    Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

bFore

アイテムを前面・背面へ移動するかを指定します。

0 : 背面に移動

0以外 : 前面に移動

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetBound` , `cwSetBoundByGrpIdx` , `cwGetBound` , `cwGetBoundByGrpIdx` ,
`cwSetVisible` , `cwSetVisibleByGrpIdx` , `cwGetVisible` ,
`cwGetVisibleByGrpIdx` , `cwSetForeground`

cwSetFixedByGrpIdx

アイテムを固定項目とするか否を設定をします。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwSetFixedByGrpIdx(  
HCWREP hCwRep,      //レポートハンドル  
LPCSTR lpGrpName,   //グループ名  
LPCSTR lpItemName,  //アイテム名  
BOOL bFixed,        //固定項目とする・しない  
INT nDigit,         //インデックスの桁数  
INT nIndex          //インデックス  
);
```

VB

```
Declare Function cwSetFixedByGrpIdx ib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal lpGrpName As String, ByVal lpItemName As String, ByVal  
bFixed As Long, ByVal nDigit As Long, ByVal nIndex As Long) As Long
```

Delphi

```
function cwSetVisibleByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; bFixed: Integer; nDigit: Integer; nIndex: Integer):  
Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

bFixed

アイテムを固定項目とするか否かを設定します。以下のいずれかの値を指定します。

0 : 固定項目としない

0以外 : 固定項目とする

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIndex

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwGetFixed` , `cwSetFixed` , `cwGetFixedByGrpIdx`

cwGetFixedByGrpIdx

アイテムを固定項目とするか否かの設定を取得します。デザイナーで、グループの複製をした時に付けたインデックス（添え字）部分を変数として扱えるようになっています。

C/C++

```
BOOL cwGetFixedByGrpIdx(  
HCWREP hCwRep,          //レポートハンドル  
LPCSTR lpGrpName,      //グループ名  
LPCSTR lpItemName,     //アイテム名  
BOOL   bFixed,         //固定項目とする・しない  
INT    nDigit,         //インデックスの桁数  
INT    nIdx            //インデックス  
);
```

VB

```
Declare Function cwGetFixedByGrpIdx Lib "cwRepEng.dll" (ByVal  
hCwRep As Long, ByVal lpGrpName As String, ByVal lpItemName As  
String, ByVal bFixed As Long, ByVal nDigit As Long, ByVal nIdx As  
Long) As Long
```

Delphi

```
function cwGetFixedByGrpIdx(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; bFixed: Integer; nDigit: Integer; nIdx: Integer):  
Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

インデックス部分を省いたグループ名へのポインタを指定します。

lpItemName

アイテム名へのポインタを指定します。

bFixed

アイテムを固定項目とするか否かの設定を格納する BOOL 型のポインタを指定します。

0 : 表示しない

0 以外 : 表示する

nDigit

インデックス部分を省いたグループ名に合成(付加)するインデックス部分の桁数を指定します。0を指定すると、桁数は可変となります。

nIdx

インデックス部分を省いたグループ名に合成(付加)するインデックスの値を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、`cwGetLastError` 関数を使います。

参照

`cwSetFixed` , `cwGetFixed` , `cwSetFixedByGrpIdx`

2.2.7 レポートの生成 / アイテムの生成・削除 関数

cwCreateReport

レポートデータを印刷エンジン内部に生成します。生成したレポートを特定するハンドルを返します。

C/C++

```
HCWREP WINAPI cwCreateReport(  
    LPCSTR      lpRepName,    //レポート名  
    INT         nMapMode,     //長さの単位  
    RECT        *pMargin,    //余白  
    CW_PAGE_DEF *pPageDef,   //用紙の設定  
    LPCSTR      lpPrinter    //プリンタ名  
);
```

VB

```
Declare Function cwCreateReport Lib "cwRepEng.dll" (ByVal lpRepName  
As String, ByVal nMapMode As Long, prMargin As CW_RECT, pPageDef As  
CW_PAGEDEF, lpPrinter As String ) As Long
```

Delphi

```
function cwCreateReport(lpRepName:PChar;nMapMode:Integer;  
prMargin :Pointer;lpPageDef:Pointer;lpPrinter:PChar ):Integer;  
stdcall;external 'cwRepEng.dll';
```

パラメータ

lpRepName

生成するレポート名を保持している、NULL で終わる文字列へのポインタを指定します。

nMapMode

レポートの長さの単位を指定します。

0 : ミリメートルを基本単位とします。

1 : インチメートルを基本単位とします。

prMargin

余白を指定します。指定する値は、レポートの長さの単位により異なります。単位がミリメートルの場合、1/100 ミリメートルを基本単位とします。インチの場合は、1 / 1000 インチを基本単位とします。

prMargin.left : 左余白

prMargin.top : 上余白

prMargin.right : 右余白

prMargin.bottom : 下余白

lpPageDef

ページ・プロパティ・データ (CW_PAGEDEF) のポインタを指定します。通常、

cwPrinterSetupDlg 関数で取得したページ定義データを指定します。

lpPrinter

プリンタ名を保持している、NULLで終わる文字列へのポインタを指定します。
デフォルトプリンタを使用する場合は""を指定します。

戻り値

関数が成功すると、対象レポートを特定するハンドルを返します。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwCloseReport

cwCreateItem

現在選択されているレイアウトページにアイテムを生成（挿入）します。

C/C++

```
BOOL WINAPI cwCreateItem(  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName,    //アイテム名  
    INT     nType,        //アイテムの種類  
    RECT    *ItemRect     //アイテムの位置  
);
```

VB

```
Declare Function cwCreateItem Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String, ByVal  
nType As Long, ItemRect As CW_RECT )As Long
```

Delphi

```
function cwCreateItem(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar; nType: Integer; ItemRect: Pointer ): Integer;  
stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

生成するアイテムのグループ名へのポインタを指定します。グループ名が無い場合は、NULLを指定します。

lpItemName

生成するアイテムの名前へのポインタを指定します。

nType

生成するアイテムの種類を指定します。

CW_ITEMTYPE_TEXT (1)	:	テキスト
CW_ITEMTYPE_IMAGE (2)	:	イメージ
CW_ITEMTYPE_BARCODE (3)	:	バーコード
CW_ITEMTYPE_LINE(4)	:	線(Line)
CW_ITEMTYPE_BOX (5)	:	枠(Box)
CW_ITEMTYPE_ELLIPSE (6)	:	円(Ellipse)

ItemRect

生成するアイテムの位置を指定します。指定する値は、レポートの長さの単位により異なります。単位がミリメートルの場合、1/100 ミリメートルを基本単位とします。インチの場合は、1 / 1000 インチを基本単位とします。

ItemRect.left : 左位置

ItemRect.top : 上位置
ItemRect.right : 右位置
ItemRect.bottom : 下位置

戻り値

関数が成功すると、0以外の値が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwDeleteItem

cwDeleteItem

現在選択されているレイアウトページのアイテムを削除します。

C/C++

```
BOOL WINAPI cwDeleteItem (  
    HCWREP hCwRep,        //レポートハンドル  
    LPCSTR lpGrpName,     //グループ名  
    LPCSTR lpItemName     //アイテム名  
);
```

VB

```
Declare Function cwDeleteItemLib "cwRepEng.dll" ( ByVal hCwRep As  
Long, ByVal lpGrpName As String, ByVal lpItemName As String ) As Long
```

Delphi

```
function cwCreateItem(hCwRep: Integer; lpGrpName: PChar;  
lpItemName: PChar ): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

lpGrpName

削除するアイテムのグループ名へのポインタを指定します。グループ名が無い場合は、NULL を指定します。

lpItemName

削除するアイテムの名前へのポインタを指定します。

戻り値

数が成功すると、0以外の値が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います

参照

cwCreateItem

2.3 印刷ジョブ・セッションの関数

cwOpenPrintJob

レポートの印刷準備をします。

C/C++

```
BOOL cwOpenPrintJob(  
    HCWREP hCwRep,    //レポートハンドル  
    LPCSTR lpJobName, //印刷ジョブ名  
    INT nFlags,       //オプション設定  
    BOOL bPreView,   //印刷プレビューの表示  
    LPCSTR lpTitle,  //印刷プレビュー画面のタイトル  
    HWND hWndParent  //親となるウインドウ  
);
```

VB

```
Declare Function cwOpenPrintJob Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal lpJobName As String, ByVal nFlags As Long, ByVal bPreView  
As Long, ByVal lpTitle As String, ByVal hWndParent As Long) As Long
```

Delphi

```
function cwOpenPrintJob(hCwRep: Integer; lpJobName: PChar;  
nFlags: Integer; bPreView: Integer; lpTitle: PChar; hWndParent:  
Integer): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

pJobName

印刷ジョブの名前を保持している、NULL で終わる文字列へのポインタを指定します。このパラメータに NULL を指定した場合、印刷ジョブ名にレポートファイル名が使用されます。

CW_PRTFLG_FILE を nFlags に設定した場合、印刷データを出力するファイル名のパスを指定します。この場合、印刷されずに、指定パスのファイルへ印刷データが書き込まれます。

nFlags

以下のオプション設定を指定します。

CW_PRTFLG_DIALOG (1) Windows 標準の印刷ダイアログボックスを表示します。プレビュー画面表示の場合は、プレビュー画面の印刷ボタンを押すと、Windows 標準の印刷ダイアログボックスを表示します。

CW_PRTFLG_STRETCH (2) 指定用紙の大きさに納まるように、自動で元の印刷イメージを拡大縮小します。

CW_PRTFLG_FILE(4)	pJobName で指定されたファイルへ印刷データを出力します。このオプションを指定した場合、他のオプションは無視され、プリンタへの出力やプレビュー画面表示は行われません。印刷データを簡易ビューアー COREPO Viewer で表示する場合は、ファイル拡張子を、必ず、CRD にする必要があります。
CW_PRTFLG_PAPERSIZE(8)	プリンタドライバのデフォルトの用紙を使用して印刷します。
CW_PRTFLG_ORIENTATION(16)	プリンタドライバのデフォルトの用紙印刷方向を使用して印刷します。
CW_PRTFLG_SOURCE(32)	プリンタドライバのデフォルトの給紙方法を使用して印刷します。

「プレビューを表示をする」にした場合、プレビュー画面の初期状態を指定します。

CW_PREVIEW_NORMAL	(0) デフォルトの状態でプレビュー画面を表示します。
CW_PREVIEW_MINIMIZED	(256) プレビュー画面を最小化して表示します。
CW_PREVIEW_MAXIMIZED	(512) プレビュー画面を最大化して表示します。

bPreView

0 以外の値を指定すると、印刷プレビュー画面を表示します。印刷プレビュー画面は、cwClosePrintJob 関数または、cwShowPreView 関数で表示されます。

pTitle

このパラメータは、bPreView に 0 以外の値が指定された場合、有効となります。印刷プレビュー画面のタイトルを保持している、NULL で終わる文字列へのポインタを指定します。このパラメータに NULL を指定した場合、印刷印刷プレビュー画面のタイトルは「印刷プレビュー」が使用されます。

hWndParent

親となるウィンドウのハンドルを指定します。ハンドルを指定すると、プレビュー画面が親ウィンドウ内に表示されます。親ウィンドウ内に表示させない場合は、NULL を指定してください。

戻り値

関数が成功すると、0 以外の値が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwOpenPrintJobEx , cwClosePrintJob , cwPrintReport,cwOpenPrintFile

cwOpenPrintJobEx

レポートの印刷準備をします。印刷部数を指定できるように拡張された関数です

C/C++

```
BOOL cwOpenPrintJob(  
    HCWREP hCwRep,    //レポートハンドル  
    LPCSTR lpJobName, //印刷ジョブ名  
    INT     nFlags,   //オプション設定  
    BOOL   bPreView, //印刷プレビューの表示  
    LPCSTR lpTitle,  //印刷プレビュー画面のタイトル  
    HWND   hWndParent, //親となるウインドウ  
    INT     nCopies   //印刷部数  
);
```

VB

```
Declare Function cwOpenPrintJob Lib "cwRepEng.dll" (ByVal hCwRep As Long, ByVal lpJobName As String, ByVal nFlags As Long, ByVal bPreView As Long, ByVal lpTitle As String, ByVal hWndParent As Long, ByVal nCopies As Long) As Long
```

Delphi

```
function cwOpenPrintJob(hCwRep: Integer; lpJobName: PChar;  
nFlags: Integer; bPreView: Integer; lpTitle: PChar; hWndParent:  
Integer; nCopies: Integer): Integer; stdcall; external  
'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

pJobName

印刷ジョブの名前を保持している、NULL で終わる文字列へのポインタを指定します。このパラメータに NULL を指定した場合、印刷ジョブ名にレポートファイル名が使用されます。

CW_PRTFLG_FILE を nFlags に設定した場合、印刷データを出力するファイル名のパスを指定します。この場合、印刷されずに、指定パスのファイルへ印刷データが書き込まれます。

nFlags

以下のオプション設定を指定します。

- CW_PRTFLG_DIALOG (1) Windows 標準の印刷ダイアログボックスを表示します。プレビュー画面表示の場合は、プレビュー画面の印刷ボタンを押すと、Windows 標準の印刷ダイアログボックスを表示します。
- CW_PRTFLG_STRETCH (2) 指定用紙の大きさに納まるように、自動で元の印刷イメージを拡大縮小します。

CW_PRTFLG_FILE(4) pJobName で指定されたファイルへ印刷データを出力します。このオプションを指定した場合、他のオプションは無視され、プリンタへの出力やプレビュー画面表示は行われません。印刷データを簡易ビューアー COREPO Viewer で表示する場合は、ファイル拡張子を、必ず、CRD にする必要があります。

CW_PRTFLG_PAPERSIZE(8) プリンタドライバのデフォルトの用紙を使用して印刷します。

CW_PRTFLG_ORIENTATION(16) プリンタドライバのデフォルトの用紙印刷方向を使用して印刷します。

CW_PRTFLG_SOURCE(32) プリンタドライバのデフォルトの給紙方法を使用して印刷します。

「プレビューを表示をする」にした場合、プレビュー画面の初期状態を指定します。

CW_PREVIEW_NORMAL (0) デフォルトの状態でプレビュー画面を表示します。
 CW_PREVIEW_MINIMIZED (256) プレビュー画面を最小化して表示します。
 CW_PREVIEW_MAXIMIZED (512) プレビュー画面を最大化して表示します。

bPreView

0 以外の値を指定すると、印刷プレビュー画面を表示します。印刷プレビュー画面は、cwClosePrintJob 関数または、cwShowPreView 関数で表示されます。

pTitle

このパラメータは、bPreView に 0 以外の値が指定された場合、有効となります。印刷プレビュー画面のタイトルを保持している、NULL で終わる文字列へのポインタを指定します。このパラメータに NULL を指定した場合、印刷印刷プレビュー画面のタイトルは「印刷プレビュー」が使用されます。

hWndParent

親となるウィンドウのハンドルを指定します。ハンドルを指定すると、プレビュー画面が親ウィンドウ内に表示されます。親ウィンドウ内に表示させない場合は、NULL を指定してください。

nCopies

印刷部数を指定します。0 が指定された場合は、1 が指定されたものとして動作します。

戻り値

関数が成功すると、0 以外の値が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwOpenPrintJob, cwClosePrintJob, cwPrintReport, cwOpenPrintFile

cwClosePrintJob

レポート印刷の終了処理を行います。また、cwOpenPrintJob でプレビューを有効にした場合、この関数が、プレビュー画面を表示します。なお、プレビュー画面が閉じられるまでこの関数から戻りません。

C/C++

```
BOOL cwClosePrintJob(  
    HCWREP hCwRep,    //レポートハンドル  
    BOOL bMODE       //印刷ジョブのキャンセル  
);
```

VB

```
Declare Function cwClosePrintJob Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal bMODE As Long) As Long
```

Delphi

```
function cwClosePrintJob(hCwRep: Integer; bMODE: Integer): Integer;  
stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

bMode

0 以外の値を指定すると、印刷を行います。(印刷データをファイル出力する場合は、印刷データファイルが作成されます。)

0 を指定すると、印刷ジョブはキャンセルされ、印刷は行われません。(印刷データをファイル出力する場合は、印刷データファイルが作成されません。)

戻り値

関数が成功すると、0 以外の値が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwOpenPrintJob , cwPrintReport , cwShowPreview

cwGetPrvhWnd

プレビューウィンドウのハンドルを取得します。

C/C++

```
BOOL cwGetPrvhWnd(  
    HCWREP hCwRep,    //レポートハンドル  
    HWND *hWnd       //プレビュー画面のハンドル  
);
```

VB

```
Declare Function cwGetPrvhWnd Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, hWnd As Long) As Long
```

Delphi

```
function cwShowPreView(hCwRep: Integer; bMODE: Integer): Integer;  
stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

hWnd

プレビュー画面のハンドルを格納する HWND 型へのポインタを指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

cwShowPreView

プレビュー画面を表示します。印刷ジョブ途中でも呼び出すことができます。
cwOpenPrintJob 関数でプレビューを有効にした場合に動作します。

C/C++

```
BOOL cwShowPreView(  
    HCWREP hCwRep, //レポートハンドル  
    BOOL bMODE     //モーダル  
);
```

VB

```
Declare Function cwShowPreView Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal bMODE As Long) As Long
```

Delphi

```
function cwShowPreView(hCwRep: Integer; bMODE: Integer): Integer;  
stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

bMode

0 以外の値を指定すると、プレビュー画面を開いたまま、呼び出し元に戻ります。

0 を指定すると、プレビュー画面が閉じるまで、呼び出し元に戻りません。

戻り値

関数が成功すると、0 以外の値が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwClosePrintJob , cwPrintReport , cwGetPrvhWnd

cwPrintReport

指定のレポートを印刷します。

C/C++

```
BOOL cwPrintReport(  
    HCWREP hCwRep,    //レポートハンドル  
    INT numFmtPage,   //印刷対象のレイアウトページ番号  
    INT numOvlPage    //オーバーレイとして使用するレイアウトページ番号  
);
```

VB

```
Declare Function cwPrintReport Lib "cwRepEng.dll" (ByVal hCwRep As  
Long, ByVal numFmtPage As Long, ByVal numOvlPage As Long) As Long
```

Delphi

```
function cwPrintReport(hCwRep: Integer; numFmtPage: Integer;  
numOvlPage: Integer): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

numFmtPage

印刷対象のレイアウトページ番号（0から9）を指定します。

1 ページ目が0、2 ページ目が1 となります。

numOvlPage

オーバーレイとして使用するレイアウトページ番号（0から9）を指定します。

numOvlPage で指定されたページが先に描画され、次に numFmtPage で指定したページで上書きします。

印刷データファイルの印刷の場合、numFmtPage・numOvlPage の値は使用されずに、印刷データファイルの印刷データが全て印刷されます。

戻り値

関数が成功すると、0 以外の値が返ります。関数が失敗すると、0 が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwOpenPrintJob , cwClosePrintJob , cwShowPreview

cwPrvWndStatus

プレビュー画面が開いているか否かの状態を取得します。

C/C++

```
BOOL cwPrvWndStatus(  
    HCWREP hCwRep    //レポートハンドル  
);
```

VB

```
Declare Function cwPrvWndStatus Lib "cwRepEng.dll" (ByVal hCwRep As  
Long) As Long
```

Delphi

```
function cwPrvWndStatus(hCwRep: Integer): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep
レポートのハンドルを指定します。

戻り値

関数が成功すると、0以外の値が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwShowPreView , cwSetPrvWndBounds , cwGetPrvWndBounds , cwClosePrvWnd ,
cwSetPrvWndPage , cwGetPrvWndPage , cwSetPrvWndScale ,
cwGetPrvWndScale , cwSetPrvWndBarStatus , cwGetPrvWndBarStatus ,
cwPrvWndDoPrint

cwSetPrvWndBounds

プレビューウィンドウ位置・サイズを設定します。

C/C++

```
BOOL cwSetPrvWndBounds(  
    HCWREP hCwRep, //レポートハンドル  
    INT     nLeft,  //左位置  
    INT     nTop,   //上位置  
    INT     nWidth, //幅  
    INT     nHeight //高さ  
);
```

VB

```
Declare Function cwSetPrvWndBounds Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal nLeft As Long, ByVal nTop As Long, ByVal nWidth As  
Long, ByVal nHeight As Long) As Long
```

Delphi

```
function cwSetPrvWndBounds(hCwRep: Integer; nLeft: Integer; nTop:  
Integer; nWidth: Integer; nHeight: Integer): Integer; stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

nLeft

プレビュー画面の左位置を指定します。

nTop

プレビュー画面の上位置を指定します。

nWidth

プレビュー画面の幅を指定します。

nHeight

プレビュー画面の高さを指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwShowPreView , cwPrvWndStatus , cwGetPrvWndBounds , cwClosePrvWnd ,
cwSetPrvWndPage , cwGetPrvWndPage , cwSetPrvWndScale ,
cwGetPrvWndScale , cwSetPrvWndBarStatus , cwGetPrvWndBarStatus
cwPrvWndDoPrint

cwGetPrvWndBounds

プレビューウインドウ位置・サイズの設定を取得します。

C/C++

```
BOOL cwGetPrvWndBounds(  
    HCWREP hCwRep, //レポートハンドル  
    INT *nLeft, //左位置  
    INT *nTop, //上位置  
    INT *nWidth, //幅  
    INT *nHeight //高さ  
);
```

VB

```
Declare Function cwGetPrvWndBounds Lib "cwRepEng.dll" (ByVal hCwRep  
As Long,nLeft As Long,nTop As Long,nWidth As Long,nHeight As  
Long )As Long
```

Delphi

```
function cwGetPrvWndBounds(hCwRep: Integer;nLeft: Pointer;  
nTop: Pointer;nWidth: Pointer;nHeight: Pointer): Integer;stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

nLeft

プレビュー画面の左位置を格納する INT 型のポインタを指定します。

nTop

プレビュー画面の上位置を格納する INT 型のポインタを指定します。

nWidth

プレビュー画面の幅を格納する INT 型のポインタを指定します。

nHeight

プレビュー画面の高さを格納する INT 型のポインタを指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwShowPreView , cwPrvWndStatus , cwSetPrvWndBounds , cwClosePrvWnd ,
cwSetPrvWndPage , cwGetPrvWndPage , cwSetPrvWndScale ,
cwGetPrvWndScale , cwSetPrvWndBarStatus, cwGetPrvWndBarStatus ,
cwPrvWndDoPrint

cwClosePrvWnd

プレビューウィンドウを閉じます。

C/C++

```
BOOL cwClosePrvWnd(  
    HCWREP hCwRep,    //レポートハンドル  
);
```

VB

```
Declare Function cwClosePrvWnd Lib "cwRepEng.dll" (ByVal hCwRep As  
Long) As Long
```

Delphi

```
function cwClosePrvWnd(hCwRep: Integer): Integer;stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep
レポートのハンドルを指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwShowPreView , cwPrvWndStatus , cwSetPrvWndBounds ,
cwGetPrvWndBounds , cwSetPrvWndPage , cwGetPrvWndPage ,
cwSetPrvWndScale , cwGetPrvWndScale , cwSetPrvWndBarStatus ,
cwGetPrvWndBarStatus , cwPrvWndDoPrint

cwSetPrvWndPage

プレビューウインドウに表示するページ番号を設定します。

C/C++

```
BOOL cwSetPrvWndPage(  
    HCWREP hCwRep, //レポートハンドル  
    INT    nPage   //ページ番号  
);
```

VB

```
Declare Function cwSetPrvWndPage Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal nPage As Long) As Long
```

Delphi

```
function cwSetPrvWndPage(hCwRep: Integer; nPage: Integer): Integer;  
stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

nPage

表示するページ番号(1 ~)を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、cwGetLastError関数を使います。

参照

cwShowPreView , cwPrvWndStatus , cwSetPrvWndBounds ,
cwGetPrvWndBounds , cwClosePrvWnd , cwGetPrvWndPage ,
cwSetPrvWndScale , cwGetPrvWndScale , cwSetPrvWndBarStatus ,
cwGetPrvWndBarStatus , cwPrvWndDoPrint

cwGetPrvWndPage

プレビューウィンドウに表示されているページ番号を取得します。

C/C++

```
BOOL cwGetPrvWndPage(  
    HCWREP hCwRep,      //レポートハンドル  
    INT *nCurPage,     //現在表示されているページ番号  
    INT *nPages        //ページ数  
);
```

VB

```
Declare Function cwGetPrvWndPage Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, nCurPage As Long, nPages As Long) As Long
```

Delphi

```
function cwGetPrvWndPage(hCwRep: Integer; nCurPage: Pointer;  
nPages: Pointer ): Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

nCurPage

表示しているページ番号を格納する INT 型へのポインタを指定します。

nPages

ページ数を格納する INT 型へのポインタを指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwShowPreView , cwPrvWndStatus , cwSetPrvWndBounds ,
cwGetPrvWndBounds , cwClosePrvWnd , cwSetPrvWndPage ,
cwSetPrvWndScale , cwGetPrvWndScale , cwSetPrvWndBarStatus ,
cwGetPrvWndBarStatus , cwPrvWndDoPrint

cwSetPrvWndScale

プレビューウィンドウに表示する倍率を設定します。

C/C++

```
BOOL cwSetPrvWndScale(  
HCWREP hCwRep, //レポートハンドル  
INT nScale //表示倍率  
);
```

VB

```
Declare Function cwSetPrvWndScale Lib "cwRepEng.dll" (ByVal hCwRep  
As Long, ByVal nScale As Long) As Long
```

Delphi

```
function cwSetPrvWndScale(hCwRep: Integer; nScale: Integer ):  
Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

nScale

表示倍率 (3 0 ~ 2 0 0) を指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwShowPreView , cwPrvWndStatus , cwSetPrvWndBounds ,
cwGetPrvWndBounds , cwClosePrvWnd , cwSetPrvWndPage ,
cwGetPrvWndPage , cwGetPrvWndScale , cwSetPrvWndBarStatus ,
cwGetPrvWndBarStatus , cwPrvWndDoPrint

cwGetPrvWndScale

プレビューウィンドウに表示する倍率を取得します。

C/C++

```
BOOL cwGetPrvWndScale(  
    HCWREP hCwRep, //レポートハンドル  
    INT *nScale //表示倍率  
);
```

VB

```
Declare Function cwGetPrvWndScale Lib "cwRepEng.dll" (ByVal hCwRep  
As Long,nScale As Long )As Long
```

Delphi

```
function cwGetPrvWndScale(hCwRep:Integer;nScale:Pointer ):  
Integer;stdcall;external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

nScale

表示倍率を格納する INT 型へのポインタを指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwShowPreView , cwPrvWndStatus , cwSetPrvWndBounds ,
cwGetPrvWndBounds , cwClosePrvWnd , cwSetPrvWndPage ,
cwGetPrvWndPage , cwSetPrvWndScale , cwSetPrvWndBarStatus ,
cwGetPrvWndBarStatus , cwPrvWndDoPrint

cwSetPrvWndBarStatus

プレビューウィンドウのツールバー・ステータスバーの表示/非表示を設定します。

C/C++

```
BOOL cwSetPrvWndBarStatus(  
    HCWREP hCwRep, //レポートハンドル  
    INT     nStatus //バーの表示/非表示  
);
```

VB

```
Declare Function cwSetPrvWndBarStatus Lib "cwRepEng.dll"(ByVal  
hCwRep As Long,ByVal nStatus As Long )As Long
```

Delphi

```
function cwSetPrvWndBarStatus(hCwRep: Integer;nStatus: Integer):  
Integer;stdcall;external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

nStatus

バーの表示非表示を設定します。以下の値の組み合わせで設定します。

CW_DISP_TOOLBAR (1) : ツールバーを表示します。

CW_DISP_STATUSBAR (2) : ステータスバーを表示します。

CW_DISP_DISABLE_PRINT(16) : 印刷ボタンを無効にします。

CW_DISP_DISABLE_CLOSE(32) : 閉じるボタンを無効にします。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwShowPreView , cwPrvWndStatus , cwSetPrvWndBounds ,
cwGetPrvWndBounds , cwClosePrvWnd , cwSetPrvWndPage ,
cwGetPrvWndPage , cwSetPrvWndScale , cwGetPrvWndScale ,
cwGetPrvWndBarStatus , cwPrvWndDoPrint

cwGetPrvWndBarStatus

プレビューウィンドウのツールバー・ステータスバーの表示/非表示の設定を取得します。

C/C++

```
BOOL cwGetPrvWndBarStatus(  
    HCWREP hCwRep,    //レポートハンドル  
    INT     *nStatus  //バーの表示/非表示  
);
```

VB

```
Declare Function cwGetPrvWndBarStatus Lib "cwRepEng.dll" (ByVal  
    hCwRep As Long, nStatus As Long) As Long
```

Delphi

```
function cwGetPrvWndBarStatus(hCwRep: Integer; nStatus: Pointer ):  
    Integer; stdcall; external 'cwRepEng.dll';
```

パラメータ

hCwRep

レポートのハンドルを指定します。

nStatus

バーの表示非表示の設定を格納する INT 型へのポインタを指定します。

CW_DISP_TOOLBAR (1) : ツールバーを表示します。

CW_DISP_STATUSBAR (2) : ステータスバーを表示します。

CW_DISP_DISABLE_PRINT(16) : 印刷ボタンを無効にします。

CW_DISP_DISABLE_CLOSE(32) : 閉じるボタンを無効にします。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwShowPreView , cwPrvWndStatus , cwSetPrvWndBounds ,
cwGetPrvWndBounds , cwClosePrvWnd , cwSetPrvWndPage ,
cwGetPrvWndPage , cwSetPrvWndScale , cwGetPrvWndScale ,
cwSetPrvWndBarStatus , cwPrvWndDoPrint

cwPrvWndDoPrint

プレビューウィンドウの印刷ボタン押下と同様に印刷を実行します。

C/C++

```
BOOL cwPrvWndDoPrint(  
    HCWREP hCwRep    //レポートハンドル  
);
```

VB

```
Declare Function cwPrvWndDoPrint Lib "cwRepEng.dll" (ByVal hCwRep  
As Long )As Long
```

Delphi

```
function cwPrvWndDoPrint(hCwRep: Integer): Integer;stdcall;  
external 'cwRepEng.dll';
```

パラメータ

hCwRep
レポートのハンドルを指定します。

戻り値

関数が成功すると、0以外の数字が返ります。関数が失敗すると、0が返ります。
エラー情報を取得するには、cwGetLastError 関数を使います。

参照

cwShowPreView , cwPrvWndStatus , cwSetPrvWndBounds ,
cwGetPrvWndBounds , cwClosePrvWnd , cwSetPrvWndPage ,
cwGetPrvWndPage , cwSetPrvWndScale , cwGetPrvWndScale ,
cwSetPrvWndBarStatus , cwGetPrvWndBarStatus

2.4 データブロック(構造体)

2.4.1 ページ・プロパティ・データブロック (構造体) 型

CW_PAGDEF

レポートの用紙サイズ・用紙の向き・給紙方法を設定する場合、CW_PAGDEF 構造体に必要なデータをセットして、cwSetPageDef 関数を呼び出します。CW_PAGDEF 構造体は以下に示すような形で定義されています。

C/C++

```
typedef struct CW_PAGE_DEF {
    INT Orientation;    //印刷方向 Windows の DEVMODE 構造体
                       //の dmOrientation と同じです。

    INT PaperSize;     //用紙種類 Windows の DEVMODE 構造体
                       //の dmPaperSize と同じです。

    INT PaperLength;   //用紙の縦方向の長さ Windows の DEVMODE
                       //構造体の dmPaperLength と同じです。

    INT PaperWidth;    //用紙の横方向の長さ Windows の DEVMODE
                       //構造体の dmPaperWidth と同じです。

    INT DefaultSource; //給紙方法 Windows の DEVMODE 構造体
                       //の dmDefaultSource と同じです。
}CW_PAGDEF;
```

VB

```
Public Type CW_PAGDEF
    Orientation As Long
    PaperSize As Long
    PaperLength As Long
    PaperWidth As Long
    DefaultSource As Long
End Type
```

Delphi

```
type CW_PAGDEF = packed record
    Orientation: Integer;
    PaperSize: Integer;
    PaperLength: Integer;
    PaperWidth: Integer;
```

```
DefaultSource: Integer;  
end;
```

パラメータ

Orientaition

用紙の向きを指定します。

縦 DMORIENT_PORTRAIT (値 1)

横 DMORIENT_LANDSCAPE (値 2)

PaperLength

用紙の縦方向の長さ (1/10mm単位) を指定します。PaperSize が 0 の場合、指定値が有効となります。

PaperWidth

用紙の横方向の長さ (1/10mm単位) PaperSize が 0 の場合、指定値が有効となります。

DefaultSource

給紙方法を指定します。Windows の DEVMODE 構造体の dmDefaultSource と同じです。詳しくは、Windows GDI を参照してください。

PaperSize

用紙サイズの種類を設定します。0 を指定した場合、PaperLength と PaperWidth で指定した用紙サイズが使用されます。PaperSize に 0 以外の値を設定した場合、PaperLength と PaperWidth の値は使用されません。Windows の DEVMODE 構造体の dmPaperSize と同じです。詳しくは、Windows GDI を参照してください。

参照

Windows GDI DEVMODE 構造体

CW_BARCODEDEF

バーコードアイテムの詳細な設定する場合、CW_BARCODEDEF 構造体に必要なデータをセットして、cwSetBarCodeProp 関数を呼び出します。CW_BARCODEDEF 構造体は以下に示すような形で定義されています。

C/C++

```
typedef struct CW_ITEM_BARCODE_DATA {
    INT      nBarKind;
    INT      nBarRatio;
    BOOL     bCheckDigit;
    COLORREF crBarColor;
    COLORREF crBackColor;
    INT      nRotate;
    INT      nFontSize;
    CHAR     FontName[64+1];
}CW_BARCODEDEF;
```

VB

```
Public Type CW_BARCODEDEF
    nBarKind As Long
    nBarRatio As Long
    bCheckDigit As Long
    crBarColor As Long
    crBackColor As Long
    nRotate As Long
    nFontSize As Long
    FontName As String *65
End Type
```

Delphi

```
type CW_BARCODEDEF =packed record
    nBarKind:      Integer;
    nBarRatio:     Integer;
    bCheckDigit:   Integer;
    crBarColor:    Integer;
    crBackColor:   Integer;
    nRotate:       Integer;
    nFontSize:     Integer;
    FontName:      array[0..64] of Char;
end;
```

パラメータ

nBarKind

バーコードの種類を指定します。以下のいずれかの値を指定します。

CW_BK_JAN13	(0)	: JAN/EAN 1 3
CW_BK_JAN8	(1)	: JAN/EAN 8
CW_BK_UPCA	(2)	: UPCA
CW_BK_UPCE	(3)	: UPCE
CW_BK_ITF	(4)	: ITF
CW_BK_MATRIX	(5)	: MATRIX 2 of 5
CW_BK_CODE39	(6)	: CODE39
CW_BK_CODE39FULL	(7)	: CODE39 FULL ASCII
CW_BK_CODE128	(8)	: CODE128
CW_BK_NW7	(9)	: NW7
CW_BK_IATA	(10)	: IATA
CW_BK_INDUSTRIAL	(11)	: INDUSTRIAL 2 of 5
CW_BK_CUSTOMER	(12)	: 郵便カスタマー

nBarRatio

細いバーと太いバーの比率 (20 ~ 30) を指定します。20 を指定した場合、細線と太線の比率は 1 : 2 になります。25 を指定すると、細線と太線の比率は 1 : 2.5 になります。

bCheckDigit

バーコードにチェックデジットが必要な場合、自動で計算してチェックデジットをコードに追加するか否かを指定します。

0 以外の値を指定すると、自動的に、必要な場合、チェックデジットをコードに追加します。0 を指定すると、チェックデジットは自動で追加されません。

crBarColor

バーの色を指定します。

crBackColor

背景色を指定します。

nRotate

表示方向を指定します。以下のいずれかの値を指定します。

0 : 0° 1 : 90° 2 : 180° 3 : 270°

nFontSize

ヒューマンリーダブル文字 (バーコードの下に印字される文字) のサイズを指定します。単位は 1/10 ポイントです。

FontName

ヒューマンリーダブル文字 (バーコードの下に印字される文字) のフォント名を指定します。

第3章

プログラミング・ガイド

この章は、コレポDLL版印刷エンジンのプログラミングの例を紹介します。

3.1 印刷エンジン・セッションの開始と終了

印刷エンジン・セッションは、印刷エンジンのオープン（`cwOpenEngine`）で始まり、印刷エンジンのクローズ（`cwCloseEngine`）で終わります。印刷エンジンのオープン（`cwOpenEngine`）を呼び出すことで、レポート印刷エンジンにアプリケーションからの要求を受け付ける準備をさせます。印刷エンジンのクローズ（`cwCloseEngine`）はレポート印刷エンジンを終了させます。

サンプル

C/C++

```
//レポート印刷エンジンを開く
if (cwOpenEngine()==FALSE ){

//エラー処理を記述します
return FALSE;
}

.
.
.

//レポート印刷エンジンを閉じる
if (cwCloseEngine()==FALSE ){
//エラー処理を記述します
return FALSE;
}
}
```

VB

```
'レポート印刷エンジンを開く
If cwOpenEngine()=0 then
'エラー処理を記述します
Exit Sub
End If

.
.
.
```

```
'レポート印刷エンジンを閉じる
If cwCloseEngine()=0 then
    'エラー処理を記述します
    Exit Sub
End If
```

Delphi

```
//レポート印刷エンジンを開く
if (cwOpenEngine()=0 )then
begin
    //エラー処理を記述します
    exit;
end;
.
.
.
//レポート印刷エンジンを閉じる
if (cwCloseEngine()=0)then
begin
    //エラー処理を記述します
    exit;
end;
```

3.2 レポートアクセス・セッションの開始と終了

レポートアクセス・セッションはレポートのオープン (cwOpenReport) で始まり、レポートのクローズ (cwCloseReport) で終わります。レポートのオープン (cwOpenReport) は、レポートデータを印刷エンジン内部に読み込み、対象レポートを特定するハンドルを呼び出し元へ返し、レポートへのアクセス準備をします。レポートのクローズ (cwCloseReport) は、レポート印刷エンジン内部に読み込んだレポートデータを破棄し、レポートへのアクセスの終了処理をします。

サンプル

C/C++

```
HCWREP hCwRep;

//レポートを開く
hCwRep =cwOpenReport(".%GrpList.crp");
if (hCwRep ==NULL ){

//エラー処理を記述します
return FALSE;

}

.
.
.

//レポートを閉じる
if (cwCloseReport(hCwRep)==FALSE ){
//エラー処理を記述します
return FALSE;
}

//http サーバー上のレポートを開く
hCwRep =cwOpenReport("http://www.com-ware.jp/GrpList.crp");
if (hCwRep ==NULL ){
//エラー処理を記述します
return FALSE;
}
}
```

VB

```
Dim hCwRep As Long

'レポートを開く

hCwRep =cwOpenReport("..crp");

If hCwRep =0 then
```

```

        'エラー処理を記述します
    Exit Sub
End If
        .
        .
        .
    'レポートを閉じる
    If cwCloseReport(hCwRep)=0 then
        'エラー処理を記述します
        Exit Sub
    End If

    'http サーバー上のレポートを開く
    hCwRep :=cwOpenReport("http://www.com-ware.jp/GrpList.crp");
    If hCwRep =0 then
        'エラー処理を記述します
        Exit Sub
    End If

```

Delphi

```

var
    hCwRep: Integer;
begin
    //レポートを開く
    hCwRep :=cwOpenReport('.*GrpList.crp' );
    if (hCwRep =0 )then
    begin
        //エラー処理を記述します
        exit;
    end;
        .
        .
        .
    //レポートを閉じる
    if (cwCloseReport(hCwRep )=0)then
    begin
        //エラー処理を記述します
        exit;
    end;

    //http サーバー上のレポートを開く
    hCwRep :=cwOpenReport('http://www.com-ware.jp/GrpList.crp' );
    if (hCwRep =0 )then
    begin

```

```
//エラー処理を記述します  
exit;  
  
end;
```

3.3 印刷ジョブ・セッションの開始と終了

印刷ジョブ・セッションは、印刷ジョブのオープン (cwOpenPrintJob) で始まり、印刷ジョブのクローズ (cwClosePrintJob) で終わります。印刷ジョブのオープン (cwOpenPrintJob) は、レポートの印刷準備をします。印刷ジョブのクローズ (cwClosePrintJob) はレポート印刷の終了処理を行います。

サンプル

C/C++

```
/印刷 JOB の開始
if (!cwOpenPrintJob(
    hCwRep,          //レポートハンドル
    "プレビューサンプル", //印刷ジョブ名
    CW_PRTFLG_DIALOG +
    CW_PREVIEW_NORMAL, //オプション設定
    true,           //印刷プレビューの表示
    "プレビューサンプル画面", //印刷プレビュー画面の
    タイトル
    NULL //プレビューを表示するウインドウ
)){
    //エラー処理を記述します
    return FALSE;
}

.
.
.

//印刷ジョブの終了
if (!cwClosePrintJob(hCwRep , true )){
    //エラー処理を記述します
    return FALSE;
}
```

VB

```
印刷 JOB の開始 プレビューを表示する。印刷設定ダイアログを表示
する
If cwOpenPrintJob(hCwRep,"プレビューサンプル", _
    CW_PRTFLG_DIALOG +CW_PREVIEW_NORMAL, _
    true,"プレビューサンプル画面",NULL)=0 then
    'エラー処理を記述します
    Exit Sub
End If

.
.
.
```

```

'印刷ジョブの終了 (プレビューが表示される)
if cwClosePrintJob(hCwRep ,1 )=0 then
    'エラー処理を記述します
    Exit Sub
End If

```

Delphi

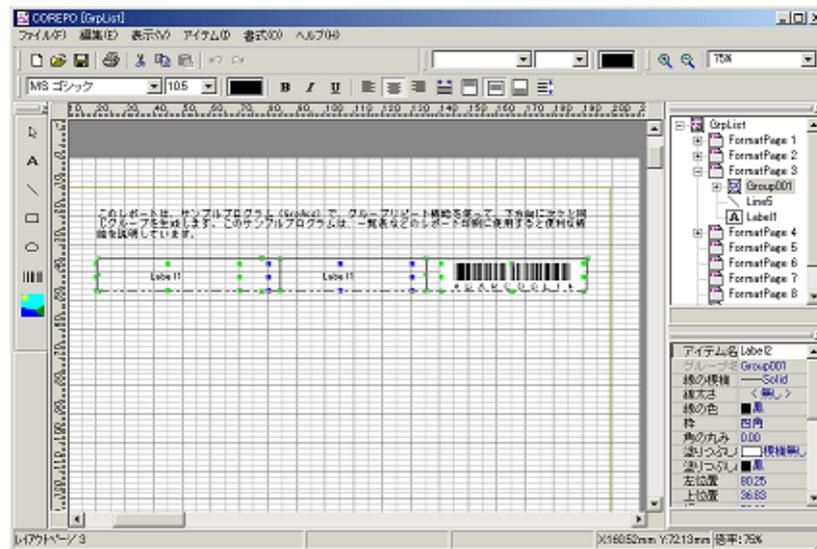
```

if (cwOpenPrintJob(
    hRep,
    'プレビューサンプル',
    CW_PRTFLG_DIALOG +CW_PREVIEW_NORMAL,
    1,
    'プレビューサンプル画面',
    0 ) = 0 )then
begin
    //エラー処理を記述します
    exit;
end;
    .
    .
    .
//印刷ジョブの終了
if (cwClosePrintJob(hRep,1 )=0 )then
begin
    //エラー処理を記述します
    exit;
end;

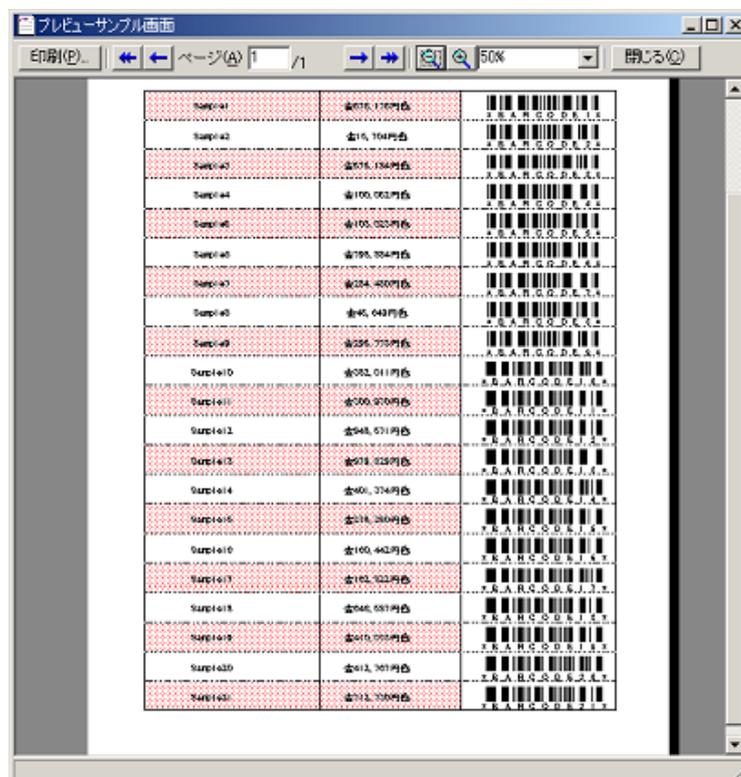
```

3.4 リポートモードでの印刷

リピート機能を使うことにより、一覧印刷などが、簡単に実現できます。たとえば、レポートでは以下のように繰り返（リピート）したいアイテムの集まりを1のグループにします。



プログラムで、リピート機能を使い、グループを下方方向に繰り返し生成して、一覧印刷帳票を作成します。



リピート機能を使うためには、印刷したいレポートのアイテムまたはグループを

cwStartRepeat 関数でリピートモードに設定します。cwStartRepeat 関数で設定できるリピートモードには、アイテムリピートモードとグループリピートモードの2種類があります。

アイテムリピートモードは、cwStartRepeat 関数でアイテム名を指定した場合に適用されるモードです。

テキストまたはバーコード アイテムがアイテムリピートモードに設定されている場合、アイテムにテキストが設定 (cwSetText/cwSetIntToStr/cwSetDoubleToStr など) されると、自動的に複製が生成されます。再びアイテムにテキストを設定すると、複製したアイテムにテキストが設定され、また新たなアイテムの複製が生成されます。但し、生成されたアイテムにテキストが設定されない限り、印刷されません。

イメージ アイテムがアイテムリピートモードに設定されている場合、アイテムにイメージが設定 (cwSetImagePath/cwSetImageDIB/cwSetDIBits など) されると、自動的に複製が生成されます。再びアイテムにイメージを設定すると、複製したアイテムにイメージが設定され、また新たなアイテムの複製が生成されます。但し、生成されたアイテムにイメージが設定されない限り、印刷されません。

線・枠 (Line/Box) アイテムがアイテムリピートモードに設定されている場合、cwDupBoxLine 関数を呼び出すことで、自動的に複製が生成されます。再び cwDupBoxLine を呼び出すと、複製したアイテムが有効となり、また新たなアイテムの複製が生成されます。但し、生成されたアイテムに対しての cwDupBoxLine 関数呼び出しが、されない限り、印刷されません。

以下にアイテムリピートモードの使い方を示します。

cwRepeat 関数でアイテムをアイテムリピートモードに設定します。

アイテムプロパティ操作関数 (cwSetImagePath/cwSetImageDIB/cwSetDIBits は省きます) で、アイテムの大きさ・フォント・色・塗りつぶし・線の種類など目的の物にします。

アイテムを複製します。

テキスト・バーコードアイテムはテキストを設定する関数 (cwSetText/cwSetIntToStr/cwSetDoubleToStr など) 呼び出すことで、複製が行われます。

線・枠・円 (Line/Box/Ellipse) アイテムでは、cwDupBoxLine 関数を呼び出すことで、複製が行われます。

イメージアイテムはイメージを設定する関数 (cwSetImagePath/cwSetImageDIB/cwSetDIBits など) 呼び出すことで、複製が行われます。

上記 から を繰り返すことで、簡単に、アイテムの複製を指定方向に指定間隔で作成できます。なお、繰り返しの結果が印刷領域を越えたか否かの確認は、`cwGetBound` または `cwCheckDupBound` 関数で行います。`cwGetBound` 関数は複製したアイテムが印刷範囲内にあるか否かを確認します。`cwCheckDupBoound` 関数は、次に複製されるアイテムが印刷範囲内に作成できるか否かを確認します。

グループリピートモードは、`cwStartRepeat` 関数でグループ名を指定し、アイテム名を `NULL` または `""` で指定した場合に適用されるモードです。

グループがグループリピートモードに設定されている場合、`cwDupGrp` 関数を呼び出すことで、自動的に複製が生成されます。再び `cwDupGrp` を呼び出すと、複製したグループが有効となり、また新たなグループの複製が生成されます。但し、生成されたグループに対して `cwDupGrp` 関数呼が呼び出されない限り、その新たに複製されたグループは印刷されません。

以下にグループリピートモードの使い方を示します。

`cwStartRepeat` 関数でグループをグループリピートモードに設定します。
(`cwStartRepeat` 関数のアイテム名を `NULL` または `""` で指定するとグループリピートモードを設定できます。)

アイテムプロパティ操作関数で、グループのアイテムのテキスト・大きさ・フォント・色・塗りつぶし・線の種類など目的の物にします。

`cwDupGrp` 関数を呼び出して、 で行ったアイテムへの変更を確定して、次のグループを複製します。

上記 から を繰り返すことで、簡単に、グループの複製を指定方向に指定間隔で作成できます。なお、繰り返しの結果印刷領域を越えたか否かの確認は、`cwGetBound` または `cwCheckDupBound` 関数で行います。`cwGetBound` 関数は複製したグループが印刷範囲内にあるか否かを確認します。`cwCheckDupBoound` 関数は、次に複製されるグループが印刷範囲内に作成できるか否かを確認します。

『サンプルにつきましては、DLL サンプルの該当言語の `GrpAcs` フォルダのソースプログラムを参考にしてください。』

3.5 印刷データのファイル出力

印刷データのファイル出力は、印刷ジョブのオープン (cwOpenPrintJob) でファイルへの出力を指定します。通常の印刷ジョブ同様、印刷ジョブのオープン (cwOpenPrintJob) 始まり、印刷ジョブのクローズ (cwClosePrintJob) で終わります。印刷ジョブのオープン (cwOpenPrintJob) は、レポートのファイル出力準備をします。印刷ジョブのクローズ (cwClosePrintJob) はレポートのファイル出力の終了処理を行います。また、cwPrintReport 関数で印刷データがファイルに出力されます。

サンプル

C/C++

```
//C:¥¥PRINTFILE¥¥Smpl.crd というファイルに印刷データを出力する
//印刷 JOB の開始
if (!cwOpenPrintJob(
    hCwRep,                //レポートハンドル
    "c:¥¥PRINTFILE¥¥Smpl.crd", //印刷データファイル名
    CW_PRTFLG_FILE,       //オプション設定
    false,
    ""
    NULL
)){
    //エラー処理を記述します
    return FALSE;
}
.
.
.
//レポートの印刷(オーバーレイ無し、レイアウトページ1ページ目)
if (!cwPrintReport(hCwRep,0,-1)){
    //エラー処理
    return FALSE;
}
.
.
.
//印刷ジョブの終了
if (!cwClosePrintJob(hCwRep ,true )){
    //エラー処理を記述します
    return FALSE;
}
```

VB

```
'印刷 JOB の開始 。C:¥PRINTFILE¥Smpl.crd というファイルに印刷データ
'を出力する
If cwOpenPrintJob(hCwRep,"c:¥PRINTFILE¥Smpl.crd", _
    CW_PRTFLG_FILE,false,"",NULL) = 0 then
    'エラー処理を記述します
    Exit Sub
End If

    .
    .
    .

'レポートの印刷(オーバーレイ無し、レイアウトページ 1 ページ目)
if cwPrintReport(hCwRep,0,-1)=0 then
    'エラー処理
    Exit Sub
EndIf

    .
    .
    .

'印刷ジョブの終了(プレビューが表示される)
if cwClosePrintJob(hCwRep,1)=0 then
    'エラー処理を記述します
    Exit Sub
End If
```

Delphi

```
//C:¥PRINTFILE¥Smpl.crd というファイルに印刷データを出力する
if (cwOpenPrintJob(
    hRep,                //レポートハンドル
    'c:¥PRINTFILE¥Smpl.crd', //印刷ジョブ名
    CW_PRTFLG_FILE,      //オプション設定
    0,
    '',
    0 ) = 0 )then
begin
    //エラー処理を記述します
    exit;
end;

    .
    .
    .

//レポートの印刷(オーバーレイ無し、レイアウトページ 1 ページ目)
if (cwPrintReport(hCwRep,0,-1)=0 )then
begin
    //エラー処理を記述します
```

```
    exit;
end;
    .
    .
    .
//印刷ジョブの終了
if (cwClosePrintJob(hRep,1 )=0 )then
begin
    //エラー処理を記述します
    exit;
end;
```

3.6 印刷データファイルの印刷

印刷データファイルの印刷は、印刷データファイルのオープン (`cwOpenPrintFile`) で始まり、レポートのクローズ (`cwCloseReport`) で終わります。印刷データファイルのオープン (`cwOpenPrintFile`) は、印刷データファイルを印刷エンジン内部に読み込み、対象レポートを特定するハンドルを呼び出し元へ返し、印刷データの印刷準備をします。レポートのクローズ (`cwCloseReport`) は、レポート印刷エンジン内部に読み込んだ印刷データを破棄し、レポートへのアクセスの終了処理をします。(注 `cwOpenPrintFile` で返されるハンドルを用いて、アイテム制御用の関数を呼び出しては意味を持ちません。)

サンプル

C/C++

```
HCWREP hCwRep;

//印刷データファイルを開く
hCwRep =cwOpenPrintFile("c:\¥¥GrpList.crd");
if (hCwRep ==NULL ){

//エラー処理を記述します
    return FALSE;
}

//印刷 JOB の開始
if (!cwOpenPrintJob(
    hCwRep,                //レポートハンドル
    "プレビューサンプル", //印刷ジョブ名
    CW_PRTFLG_DIALOG +CW_PREVIEW_NORMAL, //オプション設定
    true,                  //印刷プレビューの表示
    "プレビューサンプル画面", //印刷プレビュー画面のタイトル
    NUL                    //プレビューを表示するウインドウ
)){
//エラー処理を記述します
    return FALSE;
}

//印刷データファイルの印刷(引数のページ,オーバーレイページは意味を持ちません)
if (!cwPrintReport(hCwRep,0,-1)){
//エラー処理
    return FALSE;
}
//印刷ジョブの終了
if (!cwClosePrintJob(hCwRep ,true )){
```

```

//エラー処理を記述します
return FALSE;
}
//レポートを閉じる(印刷プレビュー)
if (!cwCloseReport(hCwRep )){
//エラー処理を記述します
return FALSE;
}

//http サーバー上の印刷データファイルを開く
hCwRep =cwOpenPrintFile("http://www.com-ware.jp/GrpList.crd");
if (hCwRep ==NULL ){
//エラー処理を記述します
return FALSE;
}
.
.

```

VB

```

Dim hCwRep As Long
'印刷データファイルを開く
hCwRep =cwOpenPrintFile("c:¥GrpList.crd");
If hCwRep =0 then
'エラー処理を記述します
Exit Sub
End If
'印刷 JOB の開始 プレビューを表示する。印刷設定ダイアログを表示
する
If cwOpenPrintJob(hCwRep,"プレビューサンプル", _
CW_PRTFLG_DIALOG +CW_PREVIEW_NORMAL,true, _
"プレビューサンプル画面",NULL)=0 then
'エラー処理を記述します
Exit Sub
End If

'印刷データファイルの印刷(引数のページ,オーバーレイページは意
味を持ちません)
if cwPrintReport(hCwRep,0,-1)=0 then
'エラー処理
Exit Sub
EndIf
'印刷ジョブの終了(プレビューが表示される)
if cwClosePrintJob(hCwRep ,1 )=0 then
'エラー処理を記述します
Exit Sub

```

```

End If
'レポートを閉じる
If cwCloseReport(hCwRep)=0 then
  'エラー処理を記述します
  Exit Sub
End If

'http サーバー上の印刷データファイルを開く
hCwRep =cwOpenPrintFile("http://www.com-ware.jp/GrpList.crp");
If hCwRep =0 then
  'エラー処理を記述します
  Exit Sub
End If
.
.
.

```

Delphi

```

var
  hCwRep: Integer;
begin

//印刷データファイルを開く
hCwRep :=cwOpenPrintFile('c:¥¥GrpList.crd' );
if (hCwRep =0 )then
begin
  //エラー処理を記述します
  exit;
end;
if (cwOpenPrintJob(
  hRep, //レポートハンドル
  'プレビューサンプル', //印刷ジョブ名
  CW_PRTFLG_DIALOG + CW_PREVIEW_NORMAL, //オプション設定
  1, //印刷プレビューの表示
  'プレビューサンプル画面', //印刷プレビュー画面のタイトル
  0 //プレビューを表示するウインドウ
) = 0 )then
begin
  //エラー処理を記述します
  exit;
end;

//印刷データファイルの印刷(引数のページ,オーバーレイページは意味を持ちません)
if (cwPrintReport(hCwRep,0,-1)=0 ) then

```

```
begin
    //エラー処理を記述します
    exit;
end;
//印刷ジョブの終了
if (cwClosePrintJob(hRep,1)=0) then
begin
    //エラー処理を記述します
    exit;
end;
//レポートを閉じる
if (cwCloseReport(hCwRep)=0) then
begin
    //エラー処理を記述します
    exit;
end;

//http サーバー上の印刷データファイルを開く
hCwRep :=cwOpenPrintFile('http://www.com-ware.jp/GrpList.crd' );
if (hCwRep =0 ) then
begin
    //エラー処理を記述します
    exit;
end;
    .
    .
    .
```

3.7 エラーモードの設定

エラーが発生したときに、印刷エンジンにエラーダイアログボックス表示を任せるか、呼び出し側のアプリケーションが処理するかを設定します。エラーモードの設定をするには、`cwSetErrorMode`関数を使います。

サンプル

C/C++

```
UINT OldMode;

//エラー発生時、レポート印刷エンジンがエラーダイアログを表示
oldMode =cwSetErrorMode(0);
.
.
.

//エラーモードを元に戻す
cwSetErrorMode(oldMode);
```

VB

```
Dim OldMode As Long

'エラー発生時、レポート印刷エンジンがエラーダイアログを表示
oldMode =cwSetErrorMode(0)
.
.
.

'エラーモードを元に戻す
cwSetErrorMode(oldMode)
```

Delphi

```
var
    IdMode: Integer;
begin

//エラー発生時、レポート印刷エンジンがエラーダイアログを表示
oldMode :=cwSetErrorMode(0);
.
.
.

//エラーモードを元に戻す
cwSetErrorMode(oldMode);
    exit;
end;
```

3.8 エラーコードの取得

エラーが発生したとき、エラー情報を取得するには、`cwGetLastError` を使います。

サンプル

C/C++

```
DWORD CrpErrCode;
DWORD WinErrCode;

//レポート印刷エンジンを開く
if (cwOpenEngine()==FALSE ){
    //エラーコードを取得
    CrpErrCode = c wGetLastError(&WinErrCode);
    if (CrpErrCode ==CRP_SYSTEM_ERROR ){
        //Windows からエラーが通知された場合の処理を記述します
        .
        .
        .
    }
    else {
        //レポート印刷エンジンのエラーが通知された場合の処理
        を記述します
        .
        .
        .
    }
}
```

VB

```
Dim CrpErrCode As Long
Dim WinErrCode As Long
'レポート印刷エンジンを開く
If cwOpenEngine()=0 then
    'エラーコードを取得
    CrpErrCode = c wGetLastError(WinErrCode)
    If CrpErrCode =CRP_SYSTEM_ERROR then
        'Windows からエラーが通知された場合の処理を記述します
        .
        .
        .
    Else
        'レポート印刷エンジンのエラーが通知された場合の処理を
        記述します
        .
    
```

```
    .  
    .  
    End If  
End If
```

Delphi

```
var  
    CrpErrCode: Integer;  
    WinErrCode: Integer;  
begin  
    .  
    .  
    //レポート印刷エンジンを開く  
    if (cwOpenEngine()=0 ) then  
    begin  
        //エラーコードを取得  
        CrpErrCode = c wGetLastError(WinErrCode)  
        if (CrpErrCode =CRP_SYSTEM_ERROR )then  
        being  
            //Windows からエラーが通知された場合の処理を記述します  
            .  
            .  
            .  
        end  
    else  
    begin  
        //レポート印刷エンジンのエラーが通知された場合の処理  
        を記述します  
        .  
        .  
        .  
    end;  
end;
```

3.9 用紙やプリンタの設定

用紙やプリンタの設定をするには、大きく分けて、2通りの方法があります。1つは、印刷時に印刷ダイアログ画面を表示して、用紙やプリンタの設定を行う方法です。もう1つは、アプリケーションで用紙やプリンタの設定ダイアログ画面を表示し、用紙やプリンタの設定内容を内部に保存して、印刷時、レポート印刷エンジンに直接指定する方法です。

3.9.1 印刷時に印刷ダイアログ画面を表示して、用紙やプリンタの設定を行う

印刷時に印刷ダイアログ画面を表示して、用紙やプリンタの設定を行うには、`cwOpenPrintJob` 関数の引数に `CW_PRTFLG_DIALOG` を指定します。詳しくは、「`cwOpenPrintJob` 関数」の説明を参照してください。

3.9.2 印刷時、レポート印刷エンジンに直接指定する

アプリケーションで用紙やプリンタの設定ダイアログ画面を表示

アプリケーションで用紙やプリンタの設定ダイアログ画面を表示するには、`cwPrinterSetupDlg` 関数を呼び出します。詳しくは、「`cwPrinterSetupDlg` 関数」の説明を参照してください。

印刷時、レポート印刷エンジンに直接指定する

印刷時、レポート印刷エンジンに直接指定するには、`cwSetPageDef` ,`cwGetPageDef` ,`cwSetPrinterName` ,`cwGetPrinterName` 関数を使います。詳しくは、それぞれ「`cwSetPageDef` 関数」,「`cwGetPageDef` 関数」,「`cwSetPrinterName` 関数」,「`cwGetPrinterName` 関数」の説明を参照してください。

サンプル

C/C++

```
CWREP      hRep;          //レポートハンドル
CW_PAGEDEF RepPageDef;  //ページプリンタ定義
char       PrinterName[32+1]; //プリンタ名
          .
          .
          .

//レポートを開く
hCwRep =cwOpenReport("Sapmle.crp");
if (hCwRep ){
    //用紙プロパティデータを取得
    if (cwGetPageDef(hCwRep ,&RepPageDef )){
        //プリンタ名を取得
```

```

        if (cwGetPrinterName(hCwRep ,PrinterName)){
            //プリンタ・用紙設定 ダイアログ画面の表示
            if (cwPrinterSetupDlg(PrinterName,&RepPageDef )){
                cwCloseReport(hCwRep);
                return TRUE;
            }
        }
    }
}
cwCloseReport(hCwRep);
return FALSE;
    .
    .
    .
//レポートを開く
hCwRep =cwOpenReport("Sapmle.crp");
if (hCwRep ){
    //用紙の設定をレポート印刷エンジンに設定する。
    if (cwSetPageDef(hCwRep ,&RepPageDef )){
        //プリンタ名をレポート印刷エンジンに設定する。
        if (cwSetPrinterName(hCwRep ,PrinterName)){
            //印刷ジョブ開始
            if (!cwOpenPrintJob(hCwRep ,
                " サンプル",0,false,"",NULL ){
                cwCloseReport(hCwRep);
                return FALSE;
            }
            .
            .
            .
            //レポート印刷エンジンのエラーが通知された場合
            の処理を記述します
            .
            .
            .
        }
    }
}
}
}

```

VB

```

Dim hRep As Long    'レポートハンドル
Dim RepPageDef As CW_PAGEDEF    'ページプリンタ定義
Dim PrinterName As String*32    'プリンタ名
    .
    .

```

```

      .
      'レポートを開く
      hCwRep =cwOpenReport("Sapmle.crp")
      if hCwRep <>0 then
        '用紙プロパティデータを取得
        if cwGetPageDef(hCwRep ,RepPageDef )<>0 then
          'プリンタ名を取得
          if cwGetPrinterName(hCwRep ,PrinterName)<>0 then
            'プリンタ・用紙設定 ダイアログ画面の表示
            if cwPrinterSetupDlg(PrinterName, _
              RepPageDef) <> 0 then
              Call cwCloseReport(hCwRep)
              Exit Sub
            End If
          End If
        End If
      End If
      Call cwCloseReport(hCwRep)
      Exit Sub
      .
      .
      .
      'レポートを開く
      hCwRep =cwOpenReport("Sapmle.crp")
      if hCwRep <>0 then
        '用紙の設定をレポート印刷エンジンに設定する。
        if cwSetPageDef(hCwRep ,RepPageDef )<>0 then
          'プリンタ名をレポート印刷エンジンに設定する。
          if cwSetPrinterName(hCwRep ,PrinterName)<>0 then
            '印刷ジョブ開始
            if cwOpenPrintJob(hCwRep ,サンプル", _
              0,false,"",NULL )=0 then
              'エラー処理
              Call cwCloseReport(hCwRep)
              Exit Sub
            End If
          End If
        End If
      End If
      .
      .
      .

```

Delphi

```

hRep: Integer;
PrinterName: array [0 ..32]of char;
RepPageDef: CW_PAGEDEF;
      .

```

```

        .
        .
//レポートを開く
hCwRep :=cwOpenReport("Sapmle.crp");
if (hCwRep <>0 )then
begin
    //用紙プロパティデータを取得
    if (cwGetPageDef(hCwRep ,Addr(RepPageDef))<>0 )then
    begin
        //プリンタ名を取得
        if (cwGetPrinterName(hCwRep ,PrinterName)<>0 )then
        begin
            //プリンタ・用紙設定 ダイアログ画面の表示
            if (cwPrinterSetupDlg(PrinterName,
                Addr(RepPageDef))<>0 )then
            begin
                cwCloseReport(hCwRep);
                exit;
            end;
        end;
    end;
end;
cwCloseReport(hCwRep);
exit;
        .
        .
        .
//レポートを開く
hCwRep :=cwOpenReport("Sapmle.crp");
if (hCwRep <>0 )then
begin
    //用紙の設定をレポート印刷エンジンに設定する。
    if (cwSetPageDef(hCwRep ,Addr(RepPageDef))<>0 )then
    begin
        //プリンタ名をレポート印刷エンジンに設定する。
        if (cwSetPrinterName(hCwRep ,PrinterName)<>0 )then
        begin
            '印刷ジョブ開始
            if (cwOpenPrintJob(hCwRep ,サンプル",
                0,false,"",NULL )=0 )then
            begin
                //エラー処理
                cwCloseReport(hCwRep);
                exit;
            end;
        end;
    end;
end;

```



3.10 文字列をテキストアイテムに設定する

テキストアイテムに文字列を設定するには、`cwSetText` 関数を使います。また、数値を文字列の数字に変換してテキストアイテムに設定するには、`cwSetIntToStr`、`cwSetUIntToStr`、`cwSetDoubleToStr` 関数を使います。詳しくは各関数の仕様を参照してください。

サンプル

C/C++

```
//文字列 " ABCD あいうえお " を設定
if (cwSetText(hCwRep,
              "",
              "Label1",
              "ABCD あいうえお")<0 ){
    //エラー処理
    return FALSE;
}

//数値を設定 全角に変換・カンマ編集有り
// 1 2 3 , 4 5 6 , 7 8 9
int nVal =123456789;
if (cwSetIntToStr(hCwRep,
                  "",
                  "Label2",
                  "",
                  "",
                  nVal,
                  CW_EDIT_ZEN +CW_EDIT_COMMA )<0 ){
    //エラー処理
    return FALSE;
}
```

VB

```
'文字列 " ABCD あいうえお " を設定
If cwSetText(hCwRep,"","Label1","ABCD あいうえお")<0 then
    'エラー処理
    Exit Sub
End If

'数値を設定 全角に変換・カンマ編集有り
' 1 2 3 , 4 5 6 , 7 8 9
Dim nVal As Long
nVal =123456789
If cwSetIntToStr(hCwRep,"","Label2","","",nVal, _
```

```

CW_EDIT_ZEN +CW_EDIT_COMMA )<0 then
  'エラー処理
  Exit Sub
End If

```

Delphi

```

//文字列 " ABCD あいうえお " を設定
if (cwSetText(hCwRep, '', 'Label1', 'ABCD あいうえお')<0 )then
begin
  //エラー処理
  exit;
end;

//数値を設定 全角に変換・カンマ編集有り
// 1 2 3 , 4 5 6 , 7 8 9
var
  nVal : Integer;
begin
  .
  .
  .
nVal :=123456789;
if (cwSetIntToStr(hCwRep,
                  '',
                  'Label2',
                  '',
                  '',
                  nVal,
                  CW_EDIT_ZEN +CW_EDIT_COMMA )<0 )then
begin
  //エラー処理
  exit;
end;

```

3.1.1 数値を文字列に変換してテキストアイテムに設定する

数値を文字列に変換してテキストアイテムに設定するには、`cwSetIntToStr`、`cwSetUIntToStr`、`cwSetDoubleToStr` 関数を使います。整数型の場合は、`cwSetIntToStr`、`cwSetUIntToStr` 関数を使います。浮動小数点型の場合は、`cwSetDoubleToStr` 関数を使います。

サンプル

C/C++

```
//数値を設定 全角に変換・カンマ編集有り
// 1 2 3 , 4 5 6 , 7 8 9
int nVal =123456789;
if (cwSetIntToStr(hCwRep,
    "",
    "Label12",
    "",
    "",
    nVal,
    CW_EDIT_ZEN +CW_EDIT_COMMA )<0 ){
    //エラー処理
    return FALSE;
}

//数値を設定 漢数字多角文字に変換
// 壹億貳千參百四拾五萬六千七百八拾九
int nVal =123456789;
if (cwSetIntToStr(hCwRep,
    "",
    "Label12",
    "",
    "",
    nVal,
    CW_EDIT_KNJ1 )<0 ){
    //エラー処理
    return FALSE;
}

//数値を設定 漢数字多角文字に変換
// 壹億貳千參百四十五萬六千七百八十九
nVal =123456789;
if (cwSetIntToStr(hCwRep,
    "",
    "Label12",
    "",
```

```

        "",
        nVal,
        CW_EDIT_KNJ2 )<0 ){
    //エラー処理
    return FALSE;
}

//数値を設定 漢数字に変換
// 一億二千三百四十五万六千七百八十九
nVal =123456789;
if (cwSetIntToStr(hCwRep,
    "",
    "Label2",
    "",
    "",
    nVal,
    CW_EDIT_KNJ3 )<0 ){
    //エラー処理
    return FALSE;
}

//数値を設定 漢数字に変換
// 一二三四五六七八九
nVal =123456789;
if (cwSetIntToStr(hCwRep,
    "",
    "Label2",
    "",
    "",
    nVal,
    CW_EDIT_KNJ4 )<0 ){
    //エラー処理
    return FALSE;
}

//浮動小数点を文字列に変換 少数点以下3桁・切り捨て
// 1 2 3 4 . 5 6 7
double dbVal =1234.56789;
if (cwSetDoubleToStr(hCwRep,
    "",
    "Label3",
    "",
    "",
    dbVal,
    3,

```

```
                                CW_EDIT_ZEN+CW_EDIT_DBL_CUT )<0 ){  
    //エラー処理  
    return FALSE;  
}
```

VB

```
Dim nVal As Long
'数値を設定 全角に変換・カンマ編集有り
' 1 2 3 , 4 5 6 , 7 8 9
nVal =123456789
If cwSetIntToStr(hCwRep,"","Label12","",",",nVal, _
    CW_EDIT_ZEN +CW_EDIT_COMMA )<0 then
    'エラー処理
    Exit Sub
End If

'数値を設定 漢数字多角文字に変換
' 壹億貳千參百四拾五萬六千七百八拾九
nVal =123456789
If cwSetIntToStr(hCwRep,"","Label12","",",",nVal, _
    CW_EDIT_KNJ1 )<0 then
    'エラー処理
    Exit Sub
End If

'数値を設定 漢数字多角文字に変換
' 壹億貳千參百四十五萬六千七百八十九
nVal =123456789
If cwSetIntToStr(hCwRep,"","Label12","",",",nVal, _
    CW_EDIT_KNJ2 )<0 then
    'エラー処理
    Exit Sub
End If

'数値を設定 漢数字に変換
' 一億二千三百四十五万六千七百八十九
nVal =123456789
If cwSetIntToStr(hCwRep,"","Label12","",",",nVal, _
    CW_EDIT_KNJ3 )<0 then
    'エラー処理
    Exit Sub
End If

'数値を設定 漢数字に変換
' 一二三四五六七八九
nVal =123456789
If cwSetIntToStr(hCwRep,"","Label12","",",",nVal, _
    CW_EDIT_KNJ4 )<0 then
    'エラー処理
    Exit Sub
```

```

End If

'浮動小数点を文字列に変換 少数点以下3桁・切り捨て
' 1 2 3 4 . 5 6 7
Dim dbVal As Double
dbVal =1234.56789
If cwSetDoubleToStr(hCwRep,"","Label3","","",dbVal, _
                    3,CW_EDIT_ZEN+CW_EDIT_DBL_CUT )<0 then
    'エラー処理
    Exit Sub
End If

```

Delphi

```

var
    nVal: Integer;
    dbVal: Double;
begin

//数値を設定 全角に変換・カンマ編集有り
// 1 2 3 , 4 5 6 , 7 8 9
nVal :=123456789;
if (cwSetIntToStr(hCwRep,
                  '',
                  'Label2',
                  '',
                  '',
                  nVal,
                  CW_EDIT_ZEN +CW_EDIT_COMMA )<0 )then
begin
    //エラー処理
    exit;
end;

//数値を設定 漢数字多角文字に変換
// 壹億貳千參百四拾五萬六千七百八拾九
nVal :=123456789;
if (cwSetIntToStr(hCwRep,
                  '',
                  'Label2',
                  '',
                  '',
                  nVal,
                  CW_EDIT_KNJ1 )<0 )then
begin
    //エラー処理

```

```

    exit;
end;

//数值を設定 漢数字多角文字に変換
// 壹億貳千參百四十五萬六千七百八十九
nVal :=123456789;
if (cwSetIntToStr(hCwRep,
    '',
    'Label12',
    '',
    '',
    nVal,
    CW_EDIT_KNJ2 )<0 ) then
begin
    //エラー処理
    exit;
end;

//数值を設定 漢数字に変換
// 一億二千三百四十五万六千七百八十九
nVal :=123456789;
if (cwSetIntToStr(hCwRep,
    '',
    'Label12',
    '',
    '',
    nVal,
    CW_EDIT_KNJ3 )<0 ) then
begin
    //エラー処理
    exit;
end;

//数值を設定 漢数字に変換
// 一二三四五六七八九
nVal :=123456789;
if (cwSetIntToStr(hCwRep,
    '',
    'Label12',
    '',
    '',
    nVal,
    CW_EDIT_KNJ4 )<0 ) then
begin
    //エラー処理

```

```

        exit;
    end;

    //浮動小数点を文字列に変換 少数点以下3桁・切り捨て
    // 1 2 3 4 . 5 6 7
    dbVal :=1234.56789;
    if (cwSetDoubleToStr(hCwRep,
                        '',
                        'Label3',
                        '',
                        '',
                        dbVal,
                        3,
                        CW_EDIT_ZEN+CW_EDIT_DBL_CUT )<0 )then
    begin
        //エラー処理
        exit;
    end;

```

3.12 文字列をバーコードアイテムに設定する

3.12.1 文字列をバーコードアイテムに設定する

文字列をバーコードアイテムに設定するには、`cwSetText` 関数を使います。詳しくは各関数の仕様を参照してください。

サンプル

C/C++

```
//文字列 " 1234567 " を設定
if ( cwSetText(hCwRep,
               "",
               "Barcode1",
               "1234567") < 0 ) {
    //エラー処理
    return FALSE;
}
```

VB

```
'文字列 " 1234567 " を設定
If cwSetText(hCwRep,"","Barcode1","1234567")<0 then
    'エラー処理
    Exit Sub
End If
```

Delphi

```
//文字列 " 1234567 " を設定
if ( cwSetText(hCwRep,'','Barcode1','1234567') < 0 ) then
begin
    //エラー処理
    exit;
end;
```

3.12.2 CODE 128に関する補足説明

CODE 128は、バーコードとそれを表すキャラクタが一對一になっているのは異なり、スタートキャラクタ、シフトキャラクタの選択により、一つのコードが3種類のデータを現せます。また、UPC CASE CODE ,EANの物流シンボル(UCC / EAN 128)として採用されています。

(1) キャラクタ構成表

指定できるキャラクタの構成表を以下に示します

数値	CODE A	CODE B	CODE C
0	[SP]	[SP]	00
1	!	!	01
2	“	“	02
3	#	#	03
4	\$	\$	04
5	%	%	05
6	&	&	06
7	‘	‘	07
8	((08
9))	09
10	*	*	10
11	+	+	11
12	,	,	12
13	-	-	13
14	.	.	14
15	/	/	15
16	0	0	16
17	1	1	17
18	2	2	18
19	3	3	19
20	4	4	20
21	5	5	21
22	6	6	22
23	7	7	23
24	8	8	24
25	9	9	25
26	:	:	26
27	;	;	27
28	<	<	28
29	=	=	29

数値	CODE A	CODE B	CODE C
54	V	V	54
55	W	W	55
56	X	X	56
57	Y	Y	57
58	Z	Z	58
59	[[1]]	59
60	¥	¥	60
61]] 1]]	61
62	^	^	62
63	_	_	63
64	[NUL]	`	64
65	[SOH]	a	65
66	[STX]	b	66
67	[ETX]	c	67
68	[EOT]	d	68
69	[ENQ]	e	69
70	[ACK]	f	70
71	[BEL]	g	71
72	[BS]	h	72
73	[HT]	i	73
74	[LF]	j	74
75	[VT]	k	75
76	[FF]	l	76
77	[CR]	m	77
78	[SO]	n	78
79	[SI]	o	79
80	[DLE]	p	80
81	[DC1]	q	81
82	[DC2]	r	82
83	[DC3]	s	83

包装	9
JANコード	4 9 0 1 2 3 4 5 6 7 8 9 4
AI (正味重量)	3 1 0 1
正味重量値	0 0 0 3 2 5

上記文字列をつなぎ合わせて、

C [F N C 1] 0 1 9 4 9 0 1 2 3 4 5 6 7 8 9 4 3 1 0 1 0 0 0 3 2 5
となります。

3 . 1 2 . 3 EAN 1 2 8に関する補足説明

EAN 1 2 8は、シンボル規格のCODE 1 2 8をベースにした物流用のアプリケーション規格です。アプリケーション識別子AIとそれに続くAIの内容を示すコードで構成されます。「AI + 内容コード」の組み合わせで梱包内容物に関する情報を現します。なお、指定できる文字は前述の「CODE 1 2 8に関する補足説明」を参照してください。AIの値や、内容コードの仕様は、UCC / EAN 1 2 8の規格書を参照してください。

(1) コード記述例

・出荷用コンテナコード (EAN 1 2 8) の記述例

包装・JANコード・正味重量の順に表現するコード記述例とするとして、包装 9、JANコード4 9 0 1 2 3 4 5 6 7 8 9 4、正味重量3 5 5であるとして、この場合の記述は、半角文字列で、

スタートコード	C
スタートキャラクタセット	[F N C 1]
AI (出荷コンテナコード)	0 1
包装	9
JANコード	4 9 0 1 2 3 4 5 6 7 8 9 4
AI (正味重量)	3 1 0 1
正味重量値	0 0 0 3 2 5

上記文字列をつなぎ合わせて、

C [F N C 1] 0 1 9 4 9 0 1 2 3 4 5 6 7 8 9 4 3 1 0 1 0 0 0 3 2 5
となります。

3.12.4 CODE 39フルアスキーに関する補足説明

CODE 39フルアスキーは、CODE 39で使用するコードを2つ組み合わせて、フルアスキーを表示することができるようにしたものです。

(1) キャラクタ構成表

指定できるキャラクタ(ASCII)の構成と、CODE 39での組み合わせを以下に示します。

CODE39 フルアスキー	CODE39	CODE39 フルアスキー	CODE39	CODE39 フルアスキー	CODE39	CODE39 フルアスキー	CODE39
[NUL]	%U	空白	空白	@	%V	`	%W
[SOH]	\$A	!	/A	A	A	a	+A
[STX]	\$B	"	/B	B	B	b	+B
[ETX]	\$C	#	/C	C	C	c	+C
[EOT]	\$D	\$	/D	D	D	d	+D
[ENQ]	\$E	%	/E	E	E	e	+E
[ACK]	\$F	&	/F	F	F	f	+F
[BEL]	\$G	'	/G	G	G	g	+G
[BS]	\$H	(/H	H	H	h	+H
[HT]	\$I)	/I	I	I	i	+I
[LF]	\$J	*	/J	J	J	j	+J
[VT]	\$K	+	/K	K	K	k	+K
[FF]	\$L	,	/L	L	L	l	+L
[CR]	\$M	-	-	M	M	m	+M
[SO]	\$N	.	.	N	N	n	+N
[SI]	\$O	/	/O	O	O	o	+O
[DLE]	\$P	0	0	P	P	p	+P
[DC1]	\$Q	1	1	Q	Q	q	+Q
[DC2]	\$R	2	2	R	R	r	+R
[DC3]	\$S	3	3	S	S	s	+S
[DC4]	\$T	4	4	T	T	t	+T
[NAK]	\$U	5	5	U	U	u	+U
[SYN]	\$V	6	6	V	V	v	+V
[ETB]	\$W	7	7	W	W	w	+W
[CAN]	\$X	8	8	X	X	x	+X
[EM]	\$Y	9	9	Y	Y	y	+Y
[SUB]	\$Z	:	/Z	Z	Z	z	+Z
[ESC]	%A	;	%F	[[1	%K	{	%P
[FS]	%B	<	%G	¥	%L		%Q
[GS]	%C	=	%H]] 1	%M	}	%R
[RS]	%D	>	%I	^	%N	~	%S

[US]	%E	?	%J	_	%O	[DEL] 2	%T
------	----	---	----	---	----	------------	----

- 1 カギ括弧 [] は制御コードの開始終了を示すため、カギ括弧を現す場合は、連続して記述します。[[と記述したカギ括弧をバーコードリードで読み込んだ場合は [] として読み込まれます。
- 2 CODE39 で、%X,%Y,%Z も CODE39 フルアスキーの DEL を現します。
- 3 制御コード (NUL,SOH,STX...) はカギ括弧で囲んで記述します。カギ括弧で囲まない場合、単なる文字列と認識されますので、必ず制御文字はカギ括弧([]) で括ってください。

(2) コード記述例

- ・最後に改行コード付加した文字列を指定する場合
“ [SAMPLE] ” という文字列に改行コードを追加して表示する場合は、
[[SAMPLE]][CR]
と記述します。

3.13 フォントの色・体裁をテキストアイテムに設定する

フォントの色・体裁をテキストアイテムに設定するには、`cwSetFont`、`cwGetFont` 関数を使います。詳しくは各関数仕様を参照してください。

サンプル

C/C++

```
char FontName; //フォント名
INT nFontStyle; //フォントスタイル
INT nFontSize; //フォントサイズ
COLORREF crFontColor; //フォントの色

//現在の設定を取得
if (cwGetFont(hCwRep,
             "",
             "Label1",
             FontName,
             &nFontStyle,
             &nFontSize,
             &crFontColor )) {
    //フォントサイズを12ポ 斜体 色を赤色にする
    nFontStyle =12*10;
    nFontStyle |=CW_FS_ITALIC;
    crFontColor =0x000000FF;
    if (cwSetFont(hCwRep,
                 "",
                 "Label1",
                 FontName,
                 &nFontStyle,
                 &nFontSize,
                 &crFontColor ) == FALSE ) {
        //エラー処理
        return FALSE;
    }
}
```

VB

```
Dim FontName As String*64 'フォント名
Dim nFontStyle As Long 'フォントスタイル
Dim nFontSize As Long 'フォントサイズ
Dim crFontColor As Long 'フォントの色

'現在の設定を取得
If cwGetFont(hCwRep,"","Label1",FontName,nFontStyle, _
```

```

        nFontSize,crFontColor)<>0 then
    'フォントサイズを 12 ポ 斜体 色を赤色にする
    nFontStyle :=12*10
    nFontStyle :=nFontStyle Or CW_FS_ITALIC
    crFontColor :=&hFF
    If cwSetFont(hCwRep,"","Label1",FontName,nFontStyle, _
        nFontSize,crFontColor )=0 then
        'エラー処理
        Exit Sub
    End If
End If

```

Delphi

```

var
    FontName:array [0 ..64]of char; //フォント名
    nFontStyle: Integer;           //フォントスタイル
    nFontSize: Integer;           //フォントサイズ
    crFontColor: Integer;         //フォントの色

begin

//現在の設定を取得
if (cwGetFont(hCwRep,
    '',
    'Label1',
    FontName,
    Addr(nFontStyle),
    Addr(nFontSize),
    Addr(crFontColor )<>0 )then
begin
    //フォントサイズを 12 ポ 斜体 色を赤色にする
    nFontStyle :=12*10;
    nFontStyle :=nFontStyle Or CW_FS_ITALIC;
    crFontColor :=$FF;
    if (cwSetFont(hCwRep,
        '',
        'Label1',
        FontName,
        Addr(nFontStyle),
        Addr(nFontSize),
        Addr(crFontColor))=0 )then
begin
    //エラー処理
    exit;
end;

```

end;

3.14 表示・非表示をアイテムに設定する

表示・非表示をアイテムに設定するには、`cwSetVisible` 関数を使います。
詳しくは `cwSetVisible` 関数仕様を参照してください。

サンプル

C/C++

```
//Lable1 アイテムを非表示にする
if (cwSetVisible(hCwRep,"","Label1",FALSE )==FALSE ){
    //エラー処理
    return FALSE;
}

//Lable2 アイテムを表示する
if (cwSetVisible(hCwRep,"","Label2",TRUE )==FALSE ){
    //エラー処理
    return FALSE;
}
```

VB

```
'Lable1 アイテムを非表示にする
If cwSetVisible(hCwRep,"","Label1",0 )=0 then
    'エラー処理
    Exit Sub
End If

'Lable2 アイテムを表示する
If cwSetVisible(hCwRep,"","Label2",1 )=0 then
    'エラー処理
    Exit Sub
End If
```

Delphi

```
//Lable1 アイテムを非表示にする
if (cwSetVisible(hCwRep,"","Label1",0 )=0 )then
begin
    //エラー処理
    exit;
end;

//Lable2 アイテムを表示する
if (cwSetVisible(hCwRep,"","Label2",1 )=0 )then
begin
    //エラー処理
```

```
    exit;  
end;
```

3.15 枠や線の色・模様，塗りつぶしの色・模様をアイテムに設定する

枠や線の色・模様をアイテムに設定するには、`cwSetPen`、`cwGetPen` 関数を使います。また、塗りつぶしの色・模様を設定するには、`cwSetBrush`、`cwGetBrush` 関数を使います。詳しくは各関数の仕様を参照してください。

サンプル

C/C++

```
int      nPs;    //線の模様
COLORREF cICI;  //線の色
int      nW;    //線の太さ
int      nR;    //角の丸み
int      nCap;  //線端の形

//線の設定を変える
//現在の設定を取得
if (cwGetPen(hRep, "", "Label1", &nPs, &cICI, &nW, &nR, &nCap)){
    //破線で、色は赤で、太さ 1.5 ポに設定
    nPs =CW_LS_DASH;
    cICI =0xff;
    nW =150;
    if (cwSetPen(hRep, "", "Label1", &nPs, &cICI,
                 &nW, &nR, &nCap)==FALSE ) {
        //エラー処理
        return FALSE;
    }
}

//枠線の設定を変える

//現在の設定を取得
if (cwGetPen(hRep, "", "Box1", &nPs, &cICI, &nW, &nR, &nCap)){
    //角に丸みをつける（半径 5mm の R）
    nR=500;
    if (cwSetPen(hRep, "", "Box1", &nPs, &cICI,
                 &nW, &nR, &nCap )==FALSE ) {
        //エラー処理
        return FALSE;
    }
}

int      nBs;    //線の模様
COLORREF cIBs;  //線の色
```

```

//塗りつぶしの設定を変える
//現在の設定を取得
if (cwGetBrush(hRep,"","Box1",&nBs,&cIBs)){
    //網目模様を設定
    nBs =CW_BS_PATTERN4;
    if (cwSetBrush(hRep,"","Box1",&nBs,&cIBs )==FALSE ){
        //エラー処理
        return FALSE;
    }
}
}

```

VB

```

Dim nPs As Long    '線の模様
Dim cICI As Long   '線の色
Dim nW As Long     '線の太さ
Dim nR As Long     '角の丸み
Dim nCap As Long   '線端の形

'線の設定を変える
'現在の設定を取得
If cwGetPen(hRep,"","Label1",nPs,cICI,nW,nR,nCap)<>0 then
    '破線で、色は赤で、太さ1.5ポに設定
    nPs =CW_LS_DASH
    cICI =&hff
    nW =150
    If cwSetPen(hRep,"","Label1",nPs,cICI,nW,nR,nCap )=0 then
        //エラー処理
        Exit Sub
    End If
End If

'枠線の設定を変える
'現在の設定を取得
If cwGetPen(hRep,"","Box1",nPs,cICI,nW,nR,nCap)<>0 then
    '角に丸みをつける(半径5mmのR)
    nR =500
    If cwSetPen(hRep,"","Box1",nPs,cICI,nW,nR,nCap )=0 then
        'エラー処理
        Exit Sub
    End If
End If

Dim nBs As Long    '線の模様
Dim cIBs As Long   '線の色

```

```

'塗りつぶしの設定を変える
'現在の設定を取得
If cwGetBrush(hRep, "", "Box1", nBs, cIBs) <> 0 then
  '網目模様を設定
  nBs := CW_BS_PATTERN4
  If cwSetBrush(hRep, "", "Box1", nBs, cIBs) = 0 then
    'エラー処理
    Exit Sub
  End If
End If
End If

```

Delphi

```

var
  nPs: Integer; //線の模様
  cICl: Integer; //線の色
  nW: Integer; //線の太さ
  nR: Integer; //角の丸み
  nCap: Integer; //線端の形
  nBs: Integer; //線の模様
  cIBs: Integer; //線の色
being

//線の設定を変える
//現在の設定を取得
if (cwGetPen(hRep, "", "Label1",
  Addr(nPs),
  Addr(cICl),
  Addr(nW),
  Addr(nR),
  Addr(nCap)) <> 0 ) then
being
  //破線で、色は赤で、太さ 1 . 5 ポに設定
  nPs := CW_LS_DASH;
  cICl := 0xff;
  nW := 150;
  if (cwSetPen(hRep, "", "Label1",
    Addr(nPs),
    Addr(cICl),
    Addr(nW),
    Addr(nR),
    Addr(nCap)) = 0 ) then
begin
  //エラー処理
  exit;
end;
end;

```

```

end;

//枠線の設定を変える
//現在の設定を取得
if (cwGetPen(hRep,"","Box1",
            Addr(nPs),
            Addr(cICl),
            Addr(nW),
            Addr(nR),
            Addr(nCap))<>0 ) then
begin
    //角に丸みをつける(半径5mmのR)
    nR :=500;
    if (cwSetPen(hRep,"","Box1",
                Addr(nPs),
                Addr(cICl),
                Addr(nW),
                Addr(nR),
                Addr(nCap))=0 ) then
        begin
            //エラー処理
            exit;
        end;
end;

//塗りつぶしの設定を変える
//現在の設定を取得
if (cwGetBrush(hRep,"","Box1",Addr(nBs),Addr(cIBs) <> 0) then
begin
    //網目模様を設定
    nBs :=CW_BS_PATTERN4;
    if (cwSetBrush(hRep,"","Box1",Addr(nBs),
                  Addr(cIBs)) = 0 ) then
        begin
            //エラー処理
            return FALSE;
        end;
end;
end;

```

3.16 イメージをイメージアイテムに設定する

イメージをイメージアイテムに設定するには、`cwSetImagePath`、`cwSetImageDIB` 関数を使います。

`cwSetImagePath` 関数は、イメージファイルをイメージアイテムに読み込ませるために使います。また、

`cwSetImageDIB` は直接、デバイス非依存型ビットマップをイメージアイテムにコピーするために使います。詳しくは各関数仕様を参照してください。

サンプル

C/C++

```
//JPG ファイル読み込んで表示させる
//イメージパスを設定
if
(cwSetImagePath(hCwRep, "", "Image1", "C:¥Test01.jpg")==FALSE )
    //エラー処理
    return FALSE;

//http サーバー上の JPG ファイル読み込んで表示させる
//イメージパスを設定
if (cwSetImagePath(hCwRep, "", "Image1",
                    "http://www.com-ware.jp/Test01.jpg")==FALSE )
    //エラー処理
    return FALSE;
```

VB

```
'JPG ファイル読み込んで表示させる
'イメージパスを設定
If cwSetImagePath(hCwRep, "", "Image1", "C:¥Test01.jpg")=0 then
    'エラー処理
    Exit Sub
End If

'http サーバー上の JPG ファイル読み込んで表示させる
'イメージパスを設定
If cwSetImagePath(hCwRep, "", "Image1", _
                  "http://www.com-ware.jp/Test01.jpg")=0 then
    'エラー処理
    Exit Sub
End If
```

Delphi

```
//JPG ファイル読み込んで表示させる
```

```
//イメージパスを設定
if (cwSetImagePath(hCwRep, '', 'Image1', 'C:¥Test01.jpg')=0 )then
begin
    //エラー処理
    exit;
end;

//http サーバー上の JPG ファイル読み込んで表示させる
//イメージパスを設定
if (cwSetImagePath(hCwRep, '', 'Image1',
                    'http://www.com-ware.jp/Test01.jpg')=0 )then
begin
    //エラー処理
    exit;
end;
```

付録

エラーコード

エラーコード	説明
括弧内は 16 進数表現	
0	(0) エラーは有りません。
1	(1) 既に印刷エンジンはオープンされています。
2	(2) 印刷エンジンはオープンされていません。
3	(3) ハンドルの値が不正です。
4	(4) グループ名又はアイテム名の指定に誤りがあります。
5	(5) 範囲を超えています。
6	(6) 指定した値はサポートされていません。
7	(7) 指定したファイルが見つかりません。
8	(8) ビットマップをコピーできません。
9	(9) 指定したパラメータに誤りがあります。
257	(101) レポートファイルを開けません。
258	(102) レポートファイルの読み込み中にエラーが発生しました。
259	(103) レポートファイルが壊れています。
260	(104) 指定したファイルはレポートファイルでは有りません。
261	(105) レポートファイルの読み込み中に CRC エラーが発生しました。
262	(106) レポートが開かれていません。
513	(201) 既に印刷ジョブは開かれています。
514	(202) 印刷ジョブは開かれていません。
515	(203) 印刷ジョブはユーザーによりキャンセルされました。
1026	(402) 既に同名のアイテムが有ります。
2049	(801) プレビューウィンドは開かれていません。
268435456 (1000000)	システムエラーが発生しました。

参照

GetLastError

索引

B

cwGetBarCodeProp, 113
cwGetBarCodePropByGrpIdx, 184
cwSetBarCodeProp, 112
cwSetBarCodePropByGrpIdx, 182
cwGetBound, 79
cwGetBoundByGrpIdx, 150
cwSetBound, 77
cwSetBoundByGrpIdx, 148
cwGetBrush, 87
cwGetBrushByGrpIdx, 158
cwSetBrush, 85
cwSetBrushByGrpIdx, 156

C

cwCheckDupBound, 129
cwClearAllBarcode, 75
cwClearAllImage, 76
cwClearAllText, 74
cwClearItemData, 73
cwClearItemDataByGrpIdx, 146
cwCloseEngine, 22
cwClosePrintJob, 209
cwClosePrvWnd, 216
cwCloseReport, 31
cwCreateItem, 202
cwCreateReport, 200
CW_BARCODEDEF, 226
CW_PAGEDEF, 224
cwSetDEVMODE, 65
cwSetDIBitsByGrpIdx, 142
cwSetDPIByGrpIdx, 144

D

cwDeleteItem, 204
cwSetDoubleToStr, 45
cwSetDoubleToStrByGrpIdx, 59

cwDupBoxLine, 127
cwDupGrp, 128

E

cwSetErrorMode, 24

F

cwGetFixed, 124
cwGetFixedByGrpIdx, 198
cwSetFixed, 123
cwSetFixedByGrpIdx, 196
cwGetFont, 91
cwGetFontByGrpIdx, 162
cwGetMultiLine, 94
cwSetFont, 89
cwSetFontByGrpIdx, 160
cwSetMultiLine, 93

G

cwGetLastError, 25
cwGetPrvWnd, 210
cwGetVersion, 23

I

cwGetImgStretch, 115
cwGetImgStretchByGrpIdx, 188
cwSetImgStretch, 114
cwSetImgStretchByGrpIdx, 186
cwSetInternetParam, 27
cwSetIntToStr, 37
cwSetIntToStrByGrpIdx, 51
cwGetItemType, 121

L

cwCloseItemHandle, 137
cwGetFirstItem, 133

cwGetItemList, 131
cwGetNextItem, 135
cwSetLongToStr, 41
cwSetLongToStrByGrpIdx, 55

O

cwSetOffset, 33
cwOpenEngine, 21
cwOpenPrintFile, 30
cwOpenPrintJob, 205
cwOpenPrintJobEx, 207
cwOpenReport, 29

P

cwGetPageDef, 62
cwSetPageDef, 61
cwGetPen, 83
cwGetPenByGrpIdx, 154
cwSetPen, 81
cwSetPenByGrpIdx, 152
cwGetPrinterName, 64
cwSetPrinterName, 63
cwPrinterSetupDlg, 26
cwPrintReport, 212
cwGetPrvWndBarStatus, 222
cwSetPrvWndBarStatus, 221
cwGetPrvWndBounds, 215
cwSetPrvWndBounds, 214
cwPrvWndDoPrint, 223
cwGetPrvWndPage, 218
cwSetPrvWndPage, 217
cwGetPrvWndScale, 220
cwSetPrvWndScale, 219
cwPrvWndStatus, 213

R

cwRenameItem, 122

S

cwSelectFmtPage, 32
cwSetDIBits, 70

cwSetDPI, 72
cwSetForeground, 120
cwSetForegroundByGrpIdx, 194
cwSetImageDIB, 68
cwSetImageDIBByGrpIdx, 140
cwSetImagePath, 66
cwSetImagePathByGrpIdx, 138
cwShowPreview, 211
cwGetSpace, 101
cwGetSpaceByGrpIdx, 170
cwSetSpace, 99
cwSetSpaceByGrpIdx, 168
cwStartRepeat, 125

T

cwGetText, 36
cwGetTextByGrpIdx, 49
cwSetText, 34
cwSetTextByGrpIdx, 47
cwGetTextAlign, 97
cwGetTextAlignByGrpIdx, 166
cwSetTextAlign, 95
cwSetTextAlignByGrpIdx, 164
cwGetTextEscapement, 104
cwGetTextEscapementByGrpIdx, 174
cwSetTextEscapement, 103
cwSetTextEscapementByGrpIdx, 172
cwGetTextMetrics, 106
cwGetTextMetricsByGrpIdx, 176

U

cwSetUIntToStr, 39
cwSetUIntToStrByGrpIdx, 53
cwSetULongToStr, 43
cwSetULongToStrByGrpIdx, 57

V

cwGetVisible, 117
cwGetVisibleByGrpIdx, 192
cwSetGrpVisible, 118
cwSetVisible, 116
cwSetVisibleByGrpIdx, 190

W

cwGetWordwrap, 110
cwGetWordwrapByGrpIdx, 180
cwSetWordwrap, 108
cwSetWordwrapByGrpIdx, 178

い

イメージをイメージアイテムに設定する, 272
印刷エンジン・セッション, 14
印刷エンジン・セッションの開始と終了, 228
印刷エンジン・セッションの関数, 15
印刷エンジン関数の一覧, 15
印刷エンジンセッションの関数, 21
印刷ジョブ・セッション, 14
印刷ジョブ・セッションの開始と終了, 233
印刷ジョブ・セッションの関数, 20
印刷データのファイル出力, 238
印刷データファイルの印刷, 241
イントロダクション, 11

え

エラーコード, 274
エラーコードの取得, 246
エラーモードの設定, 245

か

関数リファレンス, 21

こ

ごあいさつ, 7

さ

サポート, 8

す

数値を文字列に変換してテキストアイテムに
設定する, 255

と

動作環境, 9
特徴, 10

は

配布可能ファイル, 13

ひ

表示・非表示をアイテムに設定する, 266

ふ

フォントの色・体裁をテキストアイテムに設定
する, 263
付録, 274
プログラミング・ガイド, 228
プログラミングの概要, 14

へ

ページ・プロパティ・データブロック(構造体)
型, 224

も

文字列をテキストアイテムに設定する, 253
文字列をバーコードアイテムに設定する, 262

よ

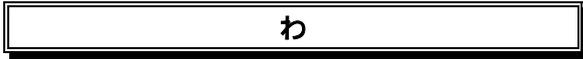
用紙やプリンタの設定, 248

り

リピートモードでの印刷, 235
レポート印刷エンジンのリンク方法, 11

れ

レポートアクセス・セッション, 14
レポートアクセス・セッションの開始と終了,
230
レポートアクセス・セッションの関数, 15



枠や線の色・模様，塗りつぶしの色・模様をアイテムに設定する, 268

帳票作成/印刷支援ツール

COREPO

D L L 版 印刷エンジンリファレンス ガイド

2 0 0 3 年 3 月 1 4 日 初版発行

2 0 0 4 年 1 0 月 1 8 日 第 6 版発行

Copyright(C) 2004 有限会社コムウェア
All rights reserved.

発 行 有限会社コムウェア

〒373-0801 太田市台之郷町 601-2

. 0276-48-0811 Fax.0276-48-0815

ファイル名 : コレポ DLLエンジン リファレンス.doc
フォルダ : C:\USERS\COREPO Help
テンプレート : C:\Documents and Settings\Hirofumi
Hamada\Application Data\Microsoft\Templates\Normal.dot
表題 : C O M W A R E
副題 :
作成者 : Hirofumi Hamada
キーワード :
説明 :
作成日時 : 2003/03/14 15:14
変更回数 : 129
最終保存日時 : 2004/10/18 11:42
最終保存者 : Hirofumi Hamada
編集時間 : 2,000 分
最終印刷日時 : 2004/10/18 18:56
最終印刷時のカウント
ページ数 : 285
単語数 : 33,005 (約)
文字数 : 188,131 (約)