

バーコード（1次元）認識ライブラリ

Version 2.0

OCX版

説明書



はじめに

本書では、バーコード（1次元）認識ライブラリ（以降、認識ライブラリと記載）の取扱方法について解説します。

- ※ Microsoft、Windows、Visual Studio は、米国その他の国における米国 Microsoft Corporation の登録商標です。
- ※ その他、記載されている社名、製品名は、一般に各社の商標または登録商標です。本書内では、®、™ 等は明記していません。

Copyright (C) 2015 Media Drive Corporation.

Copyright (C) 1991-1998, Thomas G. Lane.

This software is based in part on the work of the Independent JPEG Group.

Open Source Computer Vision Library

Copyright (C) 2000-2006, Intel Corporation, all rights reserved.

目次

1. 仕様	5
2. 動作環境	7
3. 認識対象文字	9
4. フォルダ構成	10
5. インストール	11
6. アンインストール	13
7. メンバ一覧	14
8. メンバ概要	16
9. メソッド呼び出しの流れ	17
10. メンバ説明	18
11. エラーコード一覧	65
付録A. 認識サイズと解像度	68

1. 仕様

バーコードの種類

種類	桁数	チェックデジット
JAN	8 または 13	必須： ・モジュラス 10/ウェイト 3
ITF	2～30（偶数桁のみ）	任意： ・なし（規定値） ・モジュラス 10/ウェイト 3
CODE39	3～30	任意： ・なし（規定値） ・モジュラス 43
CODE128	4～44	必須： ・モジュラス 103
NW-7	3～30	任意： ・なし（規定値） ・モジュラス 16 ・モジュラス 10/ウェイト 3 ・モジュラス 10/ウェイト 2 ・ルーンズ （モジュラス 10/ウェイト 2） ・モジュラス 11 ・加重モジュラス 11 ・7 チェック DR
カスタマバーコード	7～20	必須
GS1 DataBar Omnidirectional (RSS-14) *1	16	必須

*1 スタック（多層型）、および、合成シンボルの読み取りには対応していません。

バーコードの大きさ

コード幅	150mm 以下
コード高さ	8.0mm 以上
細バー（細スペース）*2 サイズ	2 画素以上
余白	3mm×[倍率] 以上

*2 バーコードを構成する一番細い線またはスペース

カスタマバーコードの寸法（大きさ）、印字位置（余白）、傾き

日本郵便・カスタマバーコードの仕様に従います。

日本郵便 > 郵便番号・バーコードマニュアル > バーコード

カスタマバーコードの寸法

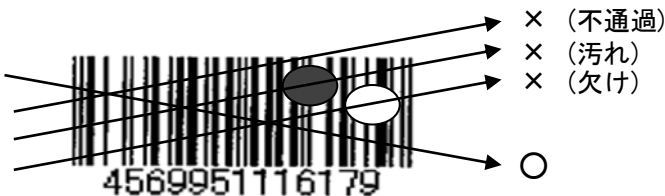
<http://www.post.japanpost.jp/zipcode/zipmanual/p12.html>

※ 拡大・縮小したサイズの異なる複数のカスタマバーコードが混在する場合は、読み取りに失敗する場合があります。

印字位置、傾き

<http://www.post.japanpost.jp/zipcode/zipmanual/p13.html>

入力／出力

入力画像形式	BMP、TIFF（非圧縮、G3/G4 圧縮、LZW 圧縮）、JPEG、PNG、PDF 白黒 2 値、グレー、カラーに対応 ※ TIFF の 1bit パレット画像には非対応 ※ マルチページ TIFF、PDF は指定ページのみ処理 ※ PDF ファイルを読み込む場合、 <ul style="list-style-type: none"> ➤ Adobe Acrobat 8 ～11 が別途必要となります。 ➤ ログイン状態が必須となります。非ログイン状態（サーバー運用）での PDF 読み込みには対応していません。 ➤ PDF の作成ツール、作成手順、内部のデータ構造によっては、PDF の読み込みに失敗する場合があります¹。
画像サイズ	スキャン画像：最大 A 3 以下（PDF 入力時は A 4 以下） 最大解像度 600dpi（300dpi 以上を推奨） カメラ画像：500 万画素以上で撮影した画像
画像の回転	90, 180, 270 度
画像の傾き	水平方向に走査したときにコードの開始から終了までを通過することが可能な角度。 
出力方式	メモリ出力 ※ JAN、ITF、CODE39、NW-7、GS1 DataBar Omnidirectional (RSS-14) については、チェックデジットをデータの一部として常に出力します。 ※ CODE39 については、スタート／ストップキャラクタ (*) をデータの一部として常に出力します。 ※ NW-7 については、スタート／ストップキャラクタ (A/B/C/D) をデータの一部として常に出力します。
その他	バーコードの色は黒色（背景は白色）を推奨

誤読や読取失敗を減らすためのヒント：

種別、桁数、読取方向を限定することで誤読や不要なコードの読み取りを減らすことができます。

また、チェックデジットを含むバーコードに対しては、チェック式を有効にすることで誤読や不要なコードの読み取りを減らすことができます。

¹ ヒント：PDF の読み込みに失敗する場合

作成ツール、手順 等を変えてお試しください。

Acrobat の場合、「印刷」メニューから「互換性のある形式」で Acrobat バージョンを個別指定して作成することにより、読取可能となることがあります。

2. 動作環境

対応 OS	<p>Windows 8.1 / Windows 8.1 Enterprise / Pro Windows 8 / Windows 8 Enterprise / Pro Windows 7 Ultimate / Enterprise / Professional / Home Premium / Starter Windows Vista Ultimate / Enterprise / Business / Home Premium / Home Basic Windows Server 2012 R2 Standard Windows Server 2012 Standard Windows Server 2008 R2 Standard / Enterprise Windows Server 2008 Standard / Enterprise 各日本語版に対応</p> <p>※ 64bit 版 Windows Windows Vista / Windows Server 2008、Windows 7 の XP モード、Macintosh は動作保証外。 ※ 64bit 版 Windows 7 / 8 / 8.1 Windows Server 2008 R2 / Windows Server 2012 / 2012 R2 では 32bit 互換モード(WOW64)で動作します。 その他の 64bit 版 OS、Windows 7 の XP モード、Macintosh は動作保 証外となります。また、Windows 8 / 8.1 / Server 2012 / 2012 R2 では デスクトップアプリケーションにのみ対応しています (Windows RT は動作保証外)。</p>
対応機種	上記 OS が正常に動作する機種
開発環境	<p>Visual C++ 2008 / 2010 / 2012 / 2013 Visual Basic 2008 / 2010 / 2012 / 2013</p>
その他	<p>本ライブラリの使用にはコピープロテクタ (USB タイプ) またはオンライン 認証 (アクティベーション) の何れかが必要。 アクティベーションを行うためには、インターネット接続環境が必要。 ※ インターネットに接続できない場合、インターネットに接続した別のパ ソコンを使って代理認証が可能。</p>

対応仮想化環境について

対応商品	<p>VMware vSphere Hypervisor(ESXi) 5.1,5.5 [ゲスト OS] Windows Server 2008 Standard/Enterprise (32bit) Windows Server 2008 R2 Standard/Enterprise Windows Server 2012 Standard Windows Server 2012 R2 Standard</p> <p>Hyper-V 3.0 [ホスト OS] Windows Server 2012 Standard [ゲスト OS] Windows Server 2008 Standard/Enterprise (32bit) Windows Server 2008 R2 Standard/Enterprise Windows Server 2012 Standard</p> <p>Hyper-V 3.1 [ホスト OS] Windows Server 2012 R2 Standard [ゲスト OS] Windows Server 2008 Standard/Enterprise (32bit) Windows Server 2008 R2 Standard/Enterprise Windows Server 2012 Standard Windows Server 2012 R2 Standard</p>
------	--

注意事項：

1. Microsoft 社からダウンロード提供されている「Microsoft Hyper-V Server 2008 R2」、「Microsoft Hyper-V Server 2012」、「Microsoft Hyper-V Server 2012 R2」はサポート対象外になります。
2. 仮想化ソフトウェアはセットアップ時の初期状態で動作確認を行っています。仮想化ソフトウェアの操作や設定についてのお問い合わせは、各メーカーのサービスセンターにお問い合わせください。
3. サポート範囲は、実環境上において発生する問題のみの対応とします。仮想化環境において個別に発生する問題、たとえば、ライブマイグレーションやバックアップなど仮想化商品固有の機能を利用した場合の動作についてはサポート対象外となります。
4. 仮想化ソフトウェアのゲスト OS 単位で、アクティベーション、または、コピープロテクタ (HASP キー) の接続 (割り当て) がそれぞれ必要になります。※Hyper-V 3.0/3.1 については、アクティベーションのみ対応 (HASP 非対応)。
5. スキャナの利用は、サポート対象外になります。

3. 認識対象文字

認識ライブラリの認識対象文字は次の通りです。

JAN	数字（0～9）のみ
ITF	数字（0～9）のみ
CODE39	数字（0～9） アルファベット大文字 記号（-, ., スペース, \$, /, +, %） スタート／ストップキャラクタ（*：アスタリスク）
NW-7	数字（0～9） 記号（-, \$, :, /, ., +） スタート／ストップキャラクタ（a～d）
CODE128	アスキーコード全文字
カスタマバーコード	数字（0～9） アルファベット大文字 記号（-） ※先頭から7桁目までは数字のみ
GS1 DataBar Omnidirectional (RSS-14)	数字（0～9）のみ

注意事項：

OCX 版のサンプルプログラムでは、読取結果をそのまま出力しています。制御コード等、表示不可能な文字を含む場合、画面表示が乱れる場合があります。読取結果の取得後、必要に応じて文字コードのチェックを別途行なってください。

4. フォルダ構成

本ライブラリの CD 構成は次の通りです。

ファイル／フォルダ	説明
Activator	
Setup.exe	ライセンス認証プログラムのインストーラ
bin	
BcDecode.exe	バーコード（1次元）認識プログラム（EXE 版）
QrDecode.exe	QRコード認識プログラム（EXE 版）
*.dll	ダイナミックリンクライブラリ（EXE/DLL/OCX 版）
*.ocx	ActiveX コントロール（OCX 版）
doc	ドキュメント
driver	
HASP	HASP ドライバ
MDCPrinter2	PDF 読み取り用仮想プリンタドライバ（32bit）
MDCPrinter2_x64	PDF 読み取り用仮想プリンタドライバ（64bit）
include	
*.h	共有ライブラリヘッダ（DLL 版）
*.lic	デザインタイムライセンスファイル（OCX 版）
lib	
*.lib	オブジェクトファイルライブラリ（DLL 版）
*.tlb	タイプライブラリ（OCX 版）
runtime	
msxml6.msi	MSXML 6.0
sample	
Image	サンプル画像
Project	サンプルプログラム（DLL/OCX 版）
SettingSample.txt	サンプル設定（EXE 版）

5. インストール

認識ライブラリ本体には、インストーラはありません。

ライブラリを構成するファイルを配置後、付属の「01_OCX のレジストリ登録.bat」を実行することで認識ライブラリ(OCX)を登録することができます。

注意事項：

Windows 8.1／8／7／Vista、Server 2012 R2／2012／2008 R2／2008 では、右クリックメニューから「管理者として実行...」を選択してください。

5.1. デザインタイムライセンスファイル

認識ライブラリを利用したアプリケーション開発には、デザインタイムライセンスファイル (MdBcDecode.lic) が必要となります。認識ライブラリ (MdBcDecode.ocx) の配置先と同一フォルダ内に MdBcDecode.lic を配置してください。

注意事項：

デザインタイムライセンスファイル (MdBcDecode.lic) の2次配布はできません。

5.2. コピープロテクタ

コピープロテクタ (ハードウェアプロテクタ、またはソフトウェアプロテクタのいずれか)
ハードウェアプロテクタ・・・USB プロテクタ
ソフトウェアプロテクタ・・・認証キー (ライセンス取得のための文字列)

(1) ハードウェアプロテクタ版の場合

本ライブラリに USB タイプのコピープロテクタが付属する場合、コピープロテクタ用の HASP デバイスドライバのインストールが必要となります。次のファイルを参照し、HASP デバイスドライバをインストールしてください。

CD 内： [driver\HASP\Readme \(HowToInstall\).txt](#)

注意事項：

HASP デバイスドライバのインストールには、管理者権限が必要となります。

(2) ソフトウェアプロテクタ版の場合

認証プログラムと認証キーを用いてオンライン認証を行うことで本ライブラリが利用可能となります。オンライン認証については、本商品に付属の「バーコード(1次元)認識ライブラリ／QR コード認識ライブラリ セットアップガイド」または次のオンラインドキュメントを参照してください。

CD 内： [doc¥ソフトウェアプロテクトの利用方法.pdf](#)

5.3. 仮想プリンタドライバ

認識ライブラリで PDF 読み込みを行う場合は、専用の仮想プリンタ (MDC Printer2) が必要となります。次の仮想プリンタドライバを必要に応じてインストールしてください。

CD 内： [driver¥MDCPrinter2¥MDCPrinter2Setup.exe](#)

注意事項：

仮想プリンタドライバのインストール時において、

- ・ MDCPrinter2Setup.exe のパスに 2 バイト文字列を含む場合、インストールに失敗する場合があります。

PDF 読み込みを行う場合、

- ・ Adobe Acrobat 8 ～11 が別途必要となります。
- ・ 処理速度が通常より低下します。
- ・ ログイン状態が必須となります。非ログイン状態（サーバー運用）での PDF 読み込みには対応していません。

6. アンインストール

付属の「02_OCX のレジストリ登録解除.bat」を実行することで、認識ライブラリ(OCX)の登録を解除することができます。登録解除後に配置したファイルを削除すれば、認識ライブラリ本体のアンインストールは完了です。

6.1. ドライバのアンインストール

CD 内の次のプログラムを用いてアンインストールを行ってください。

HASP ドライバ: `driver¥HASP¥remove.bat`

仮想プリンタドライバ: `driver¥MDCPrinter2¥MDCPrinterUnsetup.exe`

7. メンバー一覧 (OCX 版)

認識ライブラリには、27 個のメソッドと、16 個のプロパティが存在します。

7.1. メソッド一覧

【メソッド一覧】

動作	関数名
(1) バーコード認識を初期化する。	MdBcDecode_Initialize
(2) バーコード認識の終了処理をする。	MdBcDecode_Terminate
(3) 画像をファイルから読み込む。	MdBcDecode_LoadImage
(4) 画像をメモリから読み込む。	MdBcDecode_SetImage
(5) 指定領域内のバーコードを認識する。	MdBcDecode_DecodeRect
(6) 画像内のバーコードを認識する。	MdBcDecode_Decode
(7) 認識したバーコードの総数を取得する。	MdBcDecode_GetMaxCodes
(8) 認識データを取得する。	MdBcDecode_GetData
(9) 認識データサイズを取得する。	MdBcDecode_GetDataSize
(10) 認識データを文字列として取得する。	MdBcDecode_GetStrData
(11) モードを取得する。	MdBcDecode_GetMode
(12) モード情報の配列サイズを取得する。	MdBcDecode_GetModeSize
(13) バーコード認識の領域を取得する。	MdBcDecode_GetRect
(14) バーコード領域の左端座標を取得する。	MdBcDecode_GetRectLeft
(15) バーコード領域の上端座標を取得する。	MdBcDecode_GetRectTop
(16) バーコード領域の幅を取得する。	MdBcDecode_GetRectWidth
(17) バーコード領域の高さを取得する。	MdBcDecode_GetRectHeight
(18) 種別を取得する。	MdBcDecode_GetVersion
(19) 読み込んだ画像の頁数を取得する。	MdBcDecode_CountPage
(20) 読み込んだ画像のサイズを取得する。	MdBcDecode_GetImageSize
(21) 読み込んだ画像の幅を取得する。	MdBcDecode_GetImageWidth
(22) 読み込んだ画像の高さを取得する。	MdBcDecode_GetImageHeight
(23) 読み込んだ画像の横バイト数を取得する。	MdBcDecode_GetImageWidthByte
(24) 読み込んだ画像の色深度を取得する。	MdBcDecode_GetImageDepth
(25) 読み込んだ画像の横方向解像度取得する。	MdBcDecode_GetImageDpiX
(26) 読み込んだ画像の縦方向解像度取得する。	MdBcDecode_GetImageDpiY
(27) 読み込んだ画像のデータを取得する。	MdBcDecode_GetImage

7.2. プロパティ一覧

【プロパティ一覧】

動作	関数名
(1) 認識結果を取得するバーコードを選択する。	MdBcDecode_ActiveCode
(2) 認識するバーコード数の上限を設定する。	MdBcDecode_LimitCodes
(3) 仮想解像度を設定する。	MdBcDecode_VDpi
(4) 横方向の画像解像度を設定する。	MdBcDecode_XDpi
(5) 縦方向の画像解像度を設定する。	MdBcDecode_YDpi
(6) 単位長を設定する。	MdBcDecode_Unit
(7) 読取方向を設定する。	MdBcDecode_Direction
(8) 種別を設定する。	MdBcDecode_BarType
(9) 桁数を設定する。	MdBcDecode_Length
(10) ITF チェック式を設定する。	MdBcDecode_CheckITF
(11) CODE39 チェック式を設定する。	MdBcDecode_CheckCODE39
(12) NW-7 チェック式を設定する。	MdBcDecode_CheckNW7
(13) 2 値化種別を設定する。	MdBcDecode_BinType
(14) ノイズ除去サイズを設定する。	MdBcDecode_NoiseSize
(15) 穴埋めサイズを設定する。	MdBcDecode_HoleSize
(16) ITF 最小桁数を設定する。	MdBcDecode_MinITF

8. メンバ概要 (OCX 版)

8.1. メソッド概要

認識ライブラリでは、5 種類のメソッドを提供します。

- ① 認識エンジンの初期化処理、終了処理を行うメソッド
`MdBcDecode_Initialize`、`MdBcDecode_Terminate`
- ② 画像の読み込みを行うメソッド
`MdBcDecode_LoadImage`、`MdBcDecode_SetImage`、`MdBcDecode_CountPage`
- ③ 読み込んだ画像情報に関するメソッド
`MdBcDecode_GetImageSize`、`MdBcDecode_GetImageDpiX`、`MdBcDecode_GetImageDpiY`、
`MdBcDecode_GetImageWidth`、`MdBcDecode_GetImageHeight`、
`MdBcDecode_GetImageWidthByte`、`MdBcDecode_GetImageDepth`、`MdBcDecode_GetImage`
- ④ 認識を実行するメソッド
`MdBcDecode_DecodeRect`、`MdBcDecode_Decode`
- ⑤ 認識結果を取得するメソッド
`MdBcDecode_GetMaxCodes`、`MdBcDecode_GetData`、`MdBcDecode_GetDataSize`、
`MdBcDecode_GetStrData`、`MdBcDecode_GetMode`、`MdBcDecode_GetModeSize`、
`MdBcDecode_GetRect`、`MdBcDecode_GetRectLeft`、`MdBcDecode_GetRectTop`、
`MdBcDecode_GetRectWidth`、`MdBcDecode_GetRectHeight`、`MdBcDecode_GetVersion`

8.2. プロパティ概要

認識ライブラリでは、3 種類のプロパティを提供します。

- ① 情報取得対象バーコードを選択するプロパティ
`MdBcDecode_ActiveCode`
- ② 認識性能に関するプロパティ
`MdBcDecode_LimitCodes`、`MdBcDecode_Unit`、`MdBcDecode_Direction`、
`MdBcDecode_BarType`、`MdBcDecode_Length`、`MdBcDecode_CheckITF`、
`MdBcDecode_CheckCODE39`、`MdBcDecode_CheckNW7`、`MdBcDecode_BinType`、
`MdBcDecode_NoiseSize`、`MdBcDecode_HoleSize`、`MdBcDecode_MinITF`
- ③ 画像に関するプロパティ
`MdBcDecode_VDpi`、`MdBcDecode_XDpi`、`MdBcDecode_YDpi`

9. メソッド呼び出しの流れ (OCX 版)

メソッドおよびプロパティの具体的な用途は次の通りです。

9.1. 初期化・終了

認識エンジンを使用するには先ず `MdBcDecode_Initialize` で初期化を行います。
不要になった場合は、`MdBcDecode_Terminate` で終了します。

9.2. 画像読み込み

認識エンジンに画像を読み込むには、`MdBcDecode_LoadImage` を呼び出します。

9.3. 認識

認識には、`MdBcDecode_Decode` を呼び出します。

9.4. 結果取得

認識結果は、`MdBcDecode_GetStrData` で取得します。`MdBcDecode_Decode` を呼び出した結果、複数のバーコードを認識した場合は、`MdBcDecode_ActiveCode` プロパティで結果取得対象のバーコードを選択し認識結果を取得します。
バーコードに文字列以外の情報が含まれる場合や、バーコードの文字列が日本語 (S-JIS エンコーディング) 以外を想定している場合は、`MdBcDecode_GetData` で取得します。

9.5. 結果補足情報取得

結果補足情報は、`MdBcDecode_GetMode`、`MdBcDecode_GetVersion` で必要に応じて取得します。

10. メンバ説明 (OCX 版)

10.1. メソッド説明

メソッド名	1. MdBcDecode_Initialize
概要	バーコード認識を初期化する。
宣言形式	LONG MdBcDecode_Initialize()
パラメータ	なし
戻り値	LONG エラー番号
解説	バーコード認識を初期化する。 未認証で 14 日試用期間終了後、または、コピープロテクタ未挿入の場合、認識ハンドル取得に失敗してエラー:MDRC_ERR_LICENSE_PRODUCT(-32700) を返す。
使用例	<pre>// バーコード認識を初期化する ret = BcCtrl.MdBcDecode_Initialize(); if (ret != MDRC_OK) { エラー処理 ... }</pre>

メソッド名	2. MdBcDecode_Terminate
概要	バーコード認識の終了処理をする。
宣言形式	LONG MdBcDecode_Terminate()
パラメータ	なし
戻り値	LONG エラー番号
解説	バーコード認識で使用したメモリを開放する。
使用例	<pre>// バーコード認識を初期化する ret = BcCtrl.MdBcDecode_Initialize(); ... // バーコード認識の終了処理をする ret = BcCtrl.MdBcDecode_Terminate(); if (ret != MDRC_OK) { エラー処理 ... }</pre>

メソッド名	3. MdBcDecode_LoadImage				
概要	画像をファイルから読み込む。				
宣言形式	<pre>LONG MdBcDecode_LoadImage (LPCTSTR inImageFileName, LONG inPage)</pre>				
パラメータ	<table><tr><td>inImageFileName</td><td>読み込む画像ファイル名</td></tr><tr><td>inPage</td><td>読み込むページ番号</td></tr></table>	inImageFileName	読み込む画像ファイル名	inPage	読み込むページ番号
inImageFileName	読み込む画像ファイル名				
inPage	読み込むページ番号				
戻り値	<table><tr><td>LONG</td><td>エラー番号</td></tr></table>	LONG	エラー番号		
LONG	エラー番号				
解説	認識を行う画像ファイルをバーコード認識ライブラリに読み込む。				

メソッド名	4. MdBcDecode_SetImage
-------	------------------------

概要
画像をメモリから読み込む。

宣言形式
<pre> LONG MdBcDecode_SetImage (LONG* inImage, LONG inImageWidth, LONG inImageHeight, LONG inImageWidthByte) </pre>

パラメータ	
inImage	読み込む画像データのメモリアドレス
inImageWidth	読み込む画像データの横ドット数
inImageHeight	読み込む画像データの縦ドット数
inImageWidthByte	読み込む画像データの横バイト数

戻り値
LONG エラー番号

解説
<p>認識を行う画像メモリをバーコード認識ライブラリに読み込む。</p> <p>パラメータ inImage は、BYTE 型配列のメモリアドレスを LONG 型ポインタへ型変換(キャスト)を行う。(画像データは 2 値画像のみ)</p>

メソッド名	5. MdBcDecode_DecodeRect								
概要	指定領域内のバーコードを認識する。								
宣言形式	<pre>LONG MdBcDecode_DecodeRect (LONG inLeft, LONG inTop, LONG inWidth, LONG inHeight)</pre>								
パラメータ	<table><tr><td>inLeft</td><td>処理矩形左端</td></tr><tr><td>inTop</td><td>" 上端</td></tr><tr><td>inWidth</td><td>" 幅</td></tr><tr><td>inHeight</td><td>" 高さ</td></tr></table>	inLeft	処理矩形左端	inTop	" 上端	inWidth	" 幅	inHeight	" 高さ
inLeft	処理矩形左端								
inTop	" 上端								
inWidth	" 幅								
inHeight	" 高さ								
戻り値	LONG >=0 の場合は認識結果バイト数、<0 の場合はエラー番号								
解説	<p>バーコードを認識して、戻り値として認識結果バイト数を返す。</p> <p>認識結果は、MdBcDecode_GetData メソッド または、MdBcDecode_GetStrData メソッドで取得する。</p> <p>認識結果のコードの種類は、MdBcDecode_GetMode メソッド で取得する。</p> <p>バーコード画像は、3mm×[倍率] 以上の余白を必要とする。</p> <p>認識領域内で最初に見つかったコードのみを返す。</p> <p>inLeft, inTop, inWidth, inHeight で指定する領域内にバーコード以外を含む場合は、MdBcDecode_Decode 関数を使用してください。</p> <p>PDF から領域指定で読み取る場合は、処理矩形の単位をミリ単位としてください。</p>								

使用例

```
// バーコード認識を初期化する
ret = BcCtrl.MdBcDecode_Initialize();
...
// 画像を読み込む
ret = BcCtrl.MdBcDecode_LoadImage( imageFileName );

long left;      // 処理矩形左端
long top;       // "      上端
long width;     // "      幅
long height;    // "      高さ
...
// バーコードを認識する
long leng = BcCtrl.MdBcDecode_DecodeRect( left, top, width, height );
if ( leng < 0 )
{
    BcCtrl.MdBcDecode_Terminate();
    エラー処理
    ...
}
```

メソッド名	6. MdBcDecode_Decode
概要	画像内のバーコードを認識する。
宣言形式	LONG MdBcDecode_Decode()
パラメータ	なし
戻り値	LONG >=0 の場合は認識バーコード数、<0 の場合はエラー番号
解説	<p>バーコードを認識して、戻り値として認識バーコード数を返す。</p> <p>認識結果は、MdBcDecode_GetData メソッドまたは、MdBcDecode_GetStrData メソッドで取得する。</p> <p>認識結果のコードの種類は、MdBcDecode_GetMode メソッドで取得する。</p> <p>バーコード画像は、3mm×[倍率] 以上の余白を必要とする。</p> <p>画像内の複数コードを読み取り結果を返す。</p> <p>画像ページ内での読取順序は不定。</p>
使用例	<pre>// バーコード認識を初期化する ret = BcCtrl.MdBcDecode_Initialize(); ... // 画像を読み込む ret = BcCtrl.MdBcDecode_LoadImage(imageFileName); ... // バーコードを認識する mdINT32 count = BcCtrl.MdBcDecode_Decode(); if (count < 0) { BcCtrl.MdBcDecode_Terminate(); エラー処理 ... }</pre>

メソッド名	7. MdBcDecode_GetMaxCodes
概要	認識したバーコードの総数を取得する。
宣言形式	LONG MdBcDecode_GetMaxCodes ()
パラメータ	なし
戻り値	LONG 認識したバーコードの総数
解説	MdBcDecode_Decode メソッド で認識したバーコードの総数を取得する。

メソッド名	8. MdBcDecode_GetData				
概要	認識データを取得する。				
宣言形式	<pre> LONG MdBcDecode_GetData (LONG inSize, LONG* outData) </pre>				
パラメータ	<table> <tr> <td>inSize</td><td>データ配列長</td></tr> <tr> <td>outData</td><td>データ配列</td></tr> </table>	inSize	データ配列長	outData	データ配列
inSize	データ配列長				
outData	データ配列				
戻り値	<table> <tr> <td>LONG</td><td>>=0 の場合はデータをコピーしたバイト数、<0 の場合はエラー番号</td></tr> </table>	LONG	>=0 の場合はデータをコピーしたバイト数、<0 の場合はエラー番号		
LONG	>=0 の場合はデータをコピーしたバイト数、<0 の場合はエラー番号				
解説	<p>MdBcDecode_DecodeRect メソッド または、MdBcDecode_Decode メソッド で認識した結果を取得する。MdBcDecode_Decode メソッドを使用し複数のバーコードが認識された場合は、MdBcDecode_ActiveCode プロパティを使用して認識データを取得するバーコードを選択することができる。</p> <p>データ配列長は、MdBcDecode_DecodeRect を使用して認識を行った場合は、その戻り値もしくは MdBcDecode_GetDataSize メソッドの戻り値を使用し、データ配列を呼び出し側でメモリ確保する。MdBcDecode_Decode を使用して認識を行った場合は、MdBcDecode_GetDataSize メソッドの戻り値を使用し、データ配列を呼び出し側でメモリ確保する。</p> <p>データ配列のコードの種類は、MdBcDecode_GetMode メソッド で取得できるモード (配列) によって判断する。</p> <p>データ配列長は、MdBcDecode_GetMode メソッド で取得できる文字数 (配列) から求めることもできる。</p> <p>パラメータ outData は BYTE 型配列のメモリアドレスを LONG 型ポインタへ型変換 (キャスト) を行う。</p>				

使用例 1

```
// バーコードを認識する
long leng = BcCtrl.MdBcDecode_DecodeRect( left, top, width, height );
if ( leng < 0 )
{
    BcCtrl.MdBcDecode_Terminate();
    エラー処理
    ...
}

// 認識データを取得する
BYTE *data = new unsigned char[leng+1]; // データ配列
long ret = MdBcDecode_GetData( leng, (long*)data );
if ( ret != MDRC_OK )
{
    delete [] data;
    BcCtrl.MdBcDecode_Terminate();
    エラー処理
    ...
}
```

使用例 2

```
// バーコードを認識する
long count = BcCtrl.MdBcDecode_Decode();
if ( count < 0 )
{
    BcCtrl.MdBcDecode_Terminate();
    エラー処理
    ...
}

// 認識データを取得する
for ( int i = 1; i <= count; i++ )
{
    BcCtrl.SetMdBcDecode_ActiveCode(i); // バーコードを選択
    long leng = BcCtrl.MdBcDecode_GetDataSize(); // 選択したバーコードのサイズを取得
    BYTE *data = new unsigned char[leng + 1]; // データ配列
    long ret = BcCtrl.MdBcDecode_GetData(leng, (long*)data );
    if ( ret != MDRC_OK )
    {
        delete [] data;
        BcCtrl.MdBcDecode_Terminate();
        エラー発生
        ...
    }
    ...
    delete [] data
}
```

メソッド名	9. MdBcDecode_GetDataSize
概要	認識データサイズを取得する。
宣言形式	LONG MdBcDecode_GetDataSize()
パラメータ	なし
戻り値	LONG 認識結果バイト数
解説	<p>MdBcDecode_DeCodeRect メソッド または、MdBcDecode_DeCode メソッド で認識した結果のバイト数を取得する。</p> <p>MdBcDecode_DeCode メソッドを使用し複数のバーコードが認識された場合は、MdBcDecode_ActiveCode プロパティを使用して認識バイト数を取得するバーコードを選択することができる。</p>

メソッド名	10. MdBcDecode_GetStrData
概要	認識データを文字列として取得する。
宣言形式	BSTR MdBcDecode_GetStrData()
パラメータ	なし
戻り値	BSTR 認識結果文字列
解説	<p>MdBcDecode_DecodeRect メソッド または、MdBcDecode_Decode メソッド で認識した結果を文字列として取得する。</p> <p>MdBcDecode_Decode メソッドを使用し複数のバーコードが認識された場合は、MdBcDecode_ActiveCode プロパティを使用して認識データを取得するバーコードを選択することができる。</p> <p>文字列は、バーコードのデータを日本語(S-JIS)としてデコードした結果となる。</p> <p>※ JAN、ITF、CODE39、NW-7、GS1 DataBar Omnidirectional (RSS-14) については、チェックデジットをデータの一部として常に出力します。</p> <p>※ CODE39 については、スタート/ストップキャラクタ(*)をデータの一部として常に出力します。</p> <p>※ NW-7 については、スタート/ストップキャラクタ(A/B/C/D)をデータの一部として常に出力します。</p>

メソッド名	11. MdBcDecode_GetMode						
概要	モードを取得する。						
宣言形式	<pre>LONG MdBcDecode_GetMode (LONG inModeSize, LONG* outMode, LONG* outRLData)</pre>						
パラメータ	<table><tr><td>inModeSize</td><td>モード情報サイズ</td></tr><tr><td>outMode</td><td>モード情報格納メモリアドレス (NULL 指定可)</td></tr><tr><td>outRLData</td><td>モード用ランレングス情報格納メモリアドレス (NULL 指定可)</td></tr></table>	inModeSize	モード情報サイズ	outMode	モード情報格納メモリアドレス (NULL 指定可)	outRLData	モード用ランレングス情報格納メモリアドレス (NULL 指定可)
inModeSize	モード情報サイズ						
outMode	モード情報格納メモリアドレス (NULL 指定可)						
outRLData	モード用ランレングス情報格納メモリアドレス (NULL 指定可)						
戻り値	<table><tr><td>LONG</td><td>エラー番号</td></tr></table>	LONG	エラー番号				
LONG	エラー番号						
解説	<p>MdBcDecode_DecodeRect メソッド または、MdBcDecode_Decode メソッドで認識した結果のモード配列を取得する。</p> <p>MdBcDecode_Decode メソッドを使用し複数のバーコードが認識された場合は、MdBcDecode_ActiveCode プロパティを使用してモード情報を取得するバーコードを選択することができる。</p> <p>モード配列と文字数配列のサイズ(inModeSize)は MdBcDecode_GetModeSize メソッドにより取得し、呼び出し側でメモリ確保する。</p> <p>モードは認識結果のコードの種類を表し、「1/2/4/8 = 数字/英数字/8ビットバイト/漢字」* となる。バイト数/文字数はモードにより異なり、「数字/英数字/8ビットバイト/漢字 = 1/1/1/2[バイト]」† となる。本関数の使用手順を次に示す。</p> <ol style="list-style-type: none">1. MdBcDecode_GetModeSize メソッドを使用してモード配列と文字数配列のサイズを取得する。2. 1の配列数でモード配列と文字数配列のメモリを確保する。3. 2のモード配列と文字数配列を設定し、本関数を使用してモード配列と文字数配列を取得する。						

* 拡張予定の機能：現行版では、常に「1」を返す。
† 同上

使用例

```
// バーコードを認識する
long leng = BcCtrl.MdBcDecode_DecodeRect( left, top, width, height );
if ( leng < 0 )
{
    BcCtrl.MdBcDecode_Terminate();
    エラー処理
    ...
}

// モードサイズを取得する
long size = BcCtrl.MdBcDecode_GetModeSize();
long *mode = new long[size]; // モード配列
long *length = new long[size]; // 文字数配列

// モード配列と文字数配列を取得する
long ret = BcCtrl.MdBcDecode_GetMode( size, mode, length);
if ( ret != MDRC_OK )
{
    BcCtrl.MdBcDecode_Terminate();
    エラー処理
    ...
}
```

メソッド名	12. MdBcDecode_GetModeSize
概要	モード情報の配列サイズを取得する。
宣言形式	LONG MdBcDecode_GetModeSize()
パラメータ	なし
戻り値	LONG モード情報バイト数
解説	<p>MdBcDecode_DeCodeRect メソッド または、MdBcDecode_DeCode メソッドで認識した結果のモード配列サイズを取得する。</p> <p>MdBcDecode_DeCode メソッドを使用し複数のバーコードが認識された場合は、MdBcDecode_ActiveCode プロパティを使用して、モード情報配列サイズを取得するバーコードを選択することができる。</p>

メソッド名	13. MdBcDecode_GetRect
-------	------------------------

概要

バーコード認識の領域を取得する。

宣言形式

```
LONG MdBcDecode_GetRect
(
    LONG*      outLeft,
    LONG*      outTop,
    LONG*      outWidth,
    LONG*      outHeight
)
```

パラメータ

outLeft	認識矩形左端 (NULL 指定可)
outTop	" 上端 (NULL 指定可)
outWidth	" 幅 (NULL 指定可)
outHeight	" 高さ (NULL 指定可)

戻り値

LONG エラー番号

解説

認識した結果の矩形情報を取得する。
MdBcDecode_Decode メソッドを使用し複数のバーコードが認識された場合は、**MdBcDecode_ActiveCode** プロパティを使用して矩形情報を取得するバーコードを選択することができる。
 PDF から読み取り座標を取得する場合は、処理矩形の単位をミリ単位としてください。

使用例

```
// バーコードを認識する
long leng = BcCtrl.MdBcDecode_DecodeRect(
    search_left, search_top, search_width, search_height );
if ( leng < 0 )
{
    BcCtrl.MdBcDecode_Terminate();
    エラー処理
    ...
}
// 情報を取得するバーコードを選択する
BcCtrl.SetMdBcDcode_ActiveCode(1); // 1 番目のバーコードを選択

// バーコード領域を取得する
long found_left, found_top, found_width, found_height;
long ret = BcCtrl.MdBcDecode_GetRect(
    &found_left, &found_top, &found_width, &found_height );
if ( ret != MDRC_OK )
{
    BcCtrl.MdBcDecode_Terminate();
    エラー処理
    ...
}
```

メソッド名	14. MdBcDecode_GetRectLeft
概要	バーコード領域の左端座標を取得する。
宣言形式	LONG MdBcDecode_GetRectLeft()
パラメータ	なし
戻り値	LONG 認識矩形左端座標
解説	認識した結果の矩形の左端座標を取得する。 PDF から読み取り座標を取得する場合は、処理矩形の単位をミリ単位としてください。

メソッド名	15. MdBcDecode_GetRectTop
概要	バーコード領域の上端座標を取得する。
宣言形式	LONG MdBcDecode_GetRectTop()
パラメータ	なし
戻り値	LONG 認識矩形上端座標
解説	認識した結果の矩形の上端座標を取得する。 PDF から読み取り座標を取得する場合は、処理矩形の単位をミリ単位としてください。

メソッド名	16. MdBcDecode_GetRectWidth
概要	バーコード領域の幅を取得する。
宣言形式	LONG MdBcDecode_GetRectWidth()
パラメータ	なし
戻り値	LONG 認識矩形幅
解説	認識した結果の矩形の幅を取得する。 PDF から読み取り座標を取得する場合は、処理矩形の単位をミリ単位としてください。

メソッド名	17. MdBcDecode_GetRectHeight
概要	バーコード領域の高さを取得する。
宣言形式	LONG MdBcDecode_GetRectHeight()
パラメータ	なし
戻り値	LONG 認識矩形高さ
解説	認識した結果の矩形の高さを取得する。 PDF から読み取り座標を取得する場合は、処理矩形の単位をミリ単位としてください。

メソッド名	18. MdBcDecode_GetVersion
概要	<p>種別を取得する。</p> <p>種別は、JAN8/JAN13/CODE39/CODE128/ITF/NW-7/カスタマバーコード/GS1 DataBar Omnidirectional (RSS-14)</p>
宣言形式	LONG MdBcDecode_GetVersion()
パラメータ	なし
戻り値	LONG >0 の場合は種別、<=0 の場合はエラー
解説	<p>MdBcDecode_DecodeRect メソッド または、MdBcDecode_Decode メソッド で認識した結果の種別を取得する。</p> <p>MdBcDecode_Decode メソッドを使用し複数のバーコードが認識された場合は、MdBcDecode_ActiveCode プロパティを使用して種別情報を取得するバーコードを選択することができる。</p> <p>認識失敗時はエラーとなる。</p>
使用例	<pre>// バーコードを認識する long leng = BcCtrl.MdBcDecode_DecodeRect(left, top, width, height); if (leng < 0) { BcCtrl.MdBcDecode_Terminate(); エラー処理 ... } // 情報を取得するバーコードを選択する BcCtrl.SetMdBcDecode_ActiveCode(1); // 1 番目のバーコードを選択 // 種別を取得する long version = BcCtrl.MdBcDecode_GetVersion(); if (version <= 0) { BcCtrl.MdBcDecode_Terminate(); エラー処理 ... }</pre>

メソッド名	19. MdBcDecode_CountPage		
概要	読み込んだ画像の頁数を取得する。		
宣言形式	<pre> LONG MdBcDecode_CountPage (LPCTSTR inImageFileName) </pre>		
パラメータ	<table> <tr> <td>inImageFileName</td><td>読み込む画像ファイル名</td></tr> </table>	inImageFileName	読み込む画像ファイル名
inImageFileName	読み込む画像ファイル名		
戻り値	<table> <tr> <td>LONG</td><td>>=0 の場合は頁数、<0 の場合はエラー番号</td></tr> </table>	LONG	>=0 の場合は頁数、<0 の場合はエラー番号
LONG	>=0 の場合は頁数、<0 の場合はエラー番号		
解説	画像に含まれる頁数を返す。		
使用例	<pre> // 画像の頁数を取得する char file = "C:\\¥MDC_BCR_LIB01¥¥Sample¥¥Image¥¥Jan8_x2. bmp"; long pages = BcCtrl.MdBcDecode_CountPage(file); if (pages < 0) { エラー発生 ... } </pre>		

メソッド名	20. MdBcDecode_GetImageSize
-------	-----------------------------

概要
読み込んだ画像のサイズを取得する。

宣言形式
<pre> LONG MdBcDecode_GetImageSize (LONG *outImageWidth, LONG *outImageHeight, LONG *outImageWidthByte, LONG *outImageDepth) </pre>

パラメータ	
outImageWidth	画像横ドット数 (NULL 指定可)
outImageHeight	画像縦ドット数 (NULL 指定可)
outImageWidthByte	画像横バイト数 (NULL 指定可)
outImageDepth	画像ビット数 (NULL 指定可)

戻り値	
LONG	エラー番号

解説
<p>画像の横幅、高さ、横バイト数およびビット数を outImageWidth, outImageHeight, outImageWidthByte, outImageDepth にそれぞれ取得する。</p>

使用例
<pre> // 画像ファイルを読み込む char file = "C:\\¥MDC_BCR_LIB01¥¥Sample¥¥Image¥¥Jan8_x2. bmp"; long ret = BcCtrl.MdBcDecode_LoadImage(file, 1); ... // 画像サイズを取得する long binWidth; // 画像の幅 long binHeight; // 画像の高さ long binWidthByte; // 画像のバイト幅 ret = BcCtrl.MdBcDecode_GetImageSize(&binWidth, &binHeight, &binWidthByte, NULL); if (ret != MDRC_OK) { エラー処理 ... } </pre>

メソッド名	21. MdBcDecode_GetImageWidth
概要	読み込んだ画像の幅を取得する。
宣言形式	LONG MdBcDecode_GetImageWidth()
パラメータ	なし
戻り値	LONG 画像横ピクセル数
解説	画像の幅を取得する。
使用例	<pre>// 画像ファイルを読み込む char file = "C:¥¥MDC_BCR_LIB01¥¥Sample¥¥Image¥¥Jan8_x2.bmp"; long ret = BcCtrl.MdBcDecode_LoadImage(file, 1); ... // 画像横ピクセル数を取得する long binWidth; // 画像の幅 binWidth = BcCtrl.MdBcDecode_GetImageWidth(); ...</pre>

メソッド名	22. MdBcDecode_GetImageHeight
概要	読み込んだ画像の高さを取得する。
宣言形式	LONG MdBcDecode_GetImageHeight()
パラメータ	なし
戻り値	LONG 画像縦ピクセル数
解説	画像の高さを取得する。
使用例	<pre>// 画像ファイルを読み込む char file = "C:¥¥MDC_BCR_LIB01¥¥Sample¥¥Image¥¥Jan8_x2.bmp"; long ret = BcCtrl.MdBcDecode_LoadImage(file, 1); ... // 画像の縦ピクセル数を取得する long binHeight; // 画像の高さ binHeight = BcCtrl.MdBcDecode_GetImageHeight(); ...</pre>

メソッド名	23. MdBcDecode_GetImageWidthByte
概要	読み込んだ画像の横バイト数を取得する。
宣言形式	LONG MdBcDecode_GetImageWidthByte()
パラメータ	なし
戻り値	LONG 画像横バイト数
解説	画像の横バイトを取得する。
使用例	<pre>// 画像ファイルを読み込む char file = "C:¥¥MDC_BCR_LIB01¥¥Sample¥¥Image¥¥Jan8_x2.bmp"; long ret = BcCtrl.MdBcDecode_LoadImage(file, 1); ... // 画像の横バイト数を取得する long binWidthByte; // 画像のバイト幅 binWidthByte = BcCtrl.MdBcDecode_GetImageWidthByte(); ...</pre>

メソッド名	24. MdBcDecode_GetImageDepth
概要	読み込んだ画像の色深度を取得する。
宣言形式	LONG MdBcDecode_GetImageDepth()
パラメータ	なし
戻り値	LONG 画像色深度
解説	画像の色深度を取得する。
使用例	<pre>// 画像ファイルを読み込む char file = "C:¥¥MDC_BCR_LIB01¥¥Sample¥¥Image¥¥Jan8_x2.bmp"; long ret = BcCtrl.MdBcDecode_LoadImage(file, 1); ... // 画像の色深度を取得する long binDepth; // 画像の色深度 binDepth = BcCtrl.MdBcDecode_GetImageDepth(); ...</pre>

メソッド名	25. MdBcDecode_GetImageDpiX
概要	読み込んだ画像の横方向解像度を取得する。
宣言形式	LONG MdBcDecode_GetImageDpiX()
パラメータ	なし
戻り値	LONG 横解像度
解説	画像の横解像度を取得する。
使用例	<pre>// 画像ファイルを読み込む char file = "C:\\¥MDC_BCR_LIB01¥¥Sample¥¥Image¥¥Jan8_x2.bmp"; long ret = BcCtrl.MdBcDecode_LoadImage(file, 1); ... // 画像の横解像度を取得する long binDpiX; // 画像の横解像度 binDpiX = BcCtrl.MdBcDecode_GetImageDpiX(); ...</pre>

メソッド名	26. MdBcDecode_GetImageDpiY
概要	読み込んだ画像の縦方向解像度を取得する。
宣言形式	LONG MdBcDecode_GetImageDpiY()
パラメータ	なし
戻り値	LONG 縦解像度
解説	画像の縦解像度を取得する。
使用例	<pre>// 画像ファイルを読み込む char file = "C:\\¥MDC_BCR_LIB01¥¥Sample¥¥Image¥¥Jan8_x2.bmp"; long ret = BcCtrl.MdBcDecode_LoadImage(file, 1); ... // 画像の縦解像度を取得する long binDpiY; // 画像の縦解像度 binDpiY = BcCtrl.MdBcDecode_GetImageDpiY(); ...</pre>

メソッド名	27. MdBcDecode_GetImage
-------	-------------------------

概要

読み込んだ画像のデータを取得する。

宣言形式

<pre> LONG MdBcDecode_GetImage (LONG *outImage) </pre>
--

パラメータ

outImage	画像配列 (2 値画像)
----------	--------------

戻り値

LONG	エラー番号
------	-------

解説

<p>MdBcDecode_LoadImage メソッドまたは MdBcDecode_SetImage メソッドで読み込んだ画像データを outImage で指定する配列に取得(コピー)する。※取得した画像は 2 値画像</p> <p>画像配列は呼び出し側でメモリを確保する。画像配列のメモリを確保するときは、BYTE 型 (1 バイト型) で確保し、MdBcDecode_GetImage メソッドへ渡す際に LONG 型ポインタへ型変換(キャスト)を行う。</p> <p>画像配列に必要な配列サイズは、MdBcDecode_GetImageWidthByte メソッド、MdBcDecode_GetImageHeight メソッドで取得する。</p>

使用例

<pre> // 画像ファイルを読み込む char file = "C:\\¥MDC_BCR_LIB01¥¥Sample¥¥Image¥¥Jan8_x2. bmp"; long ret = BcCtrl.MdBcDecode_LoadImage(file, 1); ... // 画像サイズを取得する long binWidthByte; // 画像のバイト幅 long binHeight; // 画像の高さ binWidthByte = BcCtrl.MdBcDecode_GetImageWidthByte(); binHeight = BcCtrl.MdBcDecode_GetImageHeight(); BYTE* bw = new BYTE[binWidthByte * binHeight]; ret = BcCtrl.MdBcDecode_GetImage((long*)bw); if(ret != MDRC_OK) { delete [] bw; エラー発生 ... } </pre>
--

10.2. プロパティ説明

プロパティ名	1. MdBcDecode_ActiveCode		
概要	認識結果を取得するバーコードを選択する。		
宣言形式	<pre> LONG GetMdBcDecode_ActiveCode() void SetMdBcDecode_ActiveCode(LONG inCode) </pre>		
パラメータ	<table> <tr> <td>inCode</td><td>認識結果を取得したいバーコード番号</td></tr> </table>	inCode	認識結果を取得したいバーコード番号
inCode	認識結果を取得したいバーコード番号		
戻り値	<table> <tr> <td>LONG</td><td>認識結果取得対象(アクティブ)になっているバーコード番号</td></tr> </table>	LONG	認識結果取得対象(アクティブ)になっているバーコード番号
LONG	認識結果取得対象(アクティブ)になっているバーコード番号		
解説	<p>MdBcDecode_Decode メソッド で認識したバーコードの認識結果を取得する場合、取得したいバーコード番号の設定を本プロパティにて行う。(MdBcDecode_Decode メソッドで複数のバーコードを認識した場合は、それぞれのバーコードに番号が割り振られる。番号は 1 から始まる自然数。)</p> <p>MdBcDecode_Decode メソッドで認識したバーコードの総数は、MdBcDecode_GetMaxCodes メソッドで取得することができる。</p>		

プロパティ名	2. MdBcDecode_LimitCodes		
概要	認識するバーコード数の上限を設定する。		
宣言形式	<code>LONG GetMdBcDecode_LimitCodes()</code> <code>void SetMdBcDecode_LimitCodes (LONG inLimitCodes)</code>		
パラメータ	<table><tr><td>inLimitCodes</td><td>認識するバーコード数の上限値</td></tr></table>	inLimitCodes	認識するバーコード数の上限値
inLimitCodes	認識するバーコード数の上限値		
戻り値	<table><tr><td>LONG</td><td>認識するバーコード数の上限値</td></tr></table>	LONG	認識するバーコード数の上限値
LONG	認識するバーコード数の上限値		
解説	MdBcDecode_Decode メソッド で認識時に検出するバーコード数の上限を設定する。		

プロパティ名	3. MdBcDecode_VDpi		
概要	仮想解像度を設定する。		
宣言形式	<pre> LONG GetMdBcDecode_VDpi () void SetMdBcDecode_VDpi (LONG inVDpi) </pre>		
パラメータ	<table> <tr> <td>inVDpi</td><td>仮想解像度 (200～600)</td></tr> </table>	inVDpi	仮想解像度 (200～600)
inVDpi	仮想解像度 (200～600)		
戻り値	<table> <tr> <td>LONG</td><td>仮想解像度</td></tr> </table>	LONG	仮想解像度
LONG	仮想解像度		
解説	仮想解像度を設定する。未指定時は画像解像度となる。		

プロパティ名	4. MdBcDecode_XDpi
概要	横方向の画像解像度を設定する。
宣言形式	LONG GetMdBcDecode_XDpi () void SetMdBcDecode_XDpi (LONG inXDpi)
パラメータ	inXDpi 横方向の画像解像度 (既定値: 400)
戻り値	LONG 横方向の画像解像度
解説	横方向の画像解像度を設定する。0 以下、未設定時は、400dpi となる。

プロパティ名	5. MdBcDecode_YDpi		
概要	縦方向の画像解像度を設定する。		
宣言形式	<pre> LONG GetMdBcDecode_YDpi () void SetMdBcDecode_YDpi (LONG inYDpi) </pre>		
パラメータ	<table> <tr> <td>inYDpi</td><td>縦方向の画像解像度 (既定値: 400)</td></tr> </table>	inYDpi	縦方向の画像解像度 (既定値: 400)
inYDpi	縦方向の画像解像度 (既定値: 400)		
戻り値	<table> <tr> <td>LONG</td><td>縦方向の画像解像度</td></tr> </table>	LONG	縦方向の画像解像度
LONG	縦方向の画像解像度		
解説	縦方向の画像解像度を設定する。0 以下、未設定時は 400dpi となる。		

プロパティ名	6. MdBcDecode_Unit		
概要			
単位長を設定する。			
宣言形式			
LONG GetMdBcDecode_Unit() void SetMdBcDecode_Unit(LONG inUnit)			
パラメータ			
inUnit	単位長	0 (既定値) 1	1 pixel 0.1mm
戻り値			
LONG	単位長		
解説			
単位長を設定する。			
PDF から領域指定で読み取る場合、または、PDF から読み取り座標を取得する場合は、ミリ単位 (1) を指定ください。			

プロパティ名	7. MdBcDecode_Direction		
概要			
読取方向を設定する。			
宣言形式			
LONG GetMdBcDecode_Direction() void SetMdBcDecode_Direction(LONG indirection)			
パラメータ			
inDirection	読取方向	0x0F (既定値)	自動判定
		0x01	→方向
		0x02	←方向
		0x04	↑方向
		0x08	↓方向
戻り値			
LONG	読取方向		
解説			
読取方向を設定する。			

プロパティ名	8. MdBcDecode_BarType		
概要			
種別を設定する。			
宣言形式			
LONG GetMdBcDecode_BarType()			
void SetMdBcDecode_BarType(LONG inBarType)			
パラメータ			
inBarType	種別	0xFFFFFFFF (既定値)	自動判定 (カスタムバーコード 以外)
		0x00000001	JAN13
		0x00000002	JAN8
		0x00000010	ITF
		0x00000100	NW-7
		0x00000200	CODE39
		0x00000400	CODE128
		0x00001000	カスタムバーコード 単独指定のみ有効
		0x00010000	GS1 DataBar Omnidirectional (RSS-14)
戻り値			
LONG	種別		
解説			
種別を設定する。			

プロパティ名	9. MdBcDecode_Length		
概要			
桁数を設定する。			
宣言形式			
LONG GetMdBcDecode_Length() void SetMdBcDecode_Length(LONG inLength)			
パラメータ			
inLength	桁数	0 (既定値)	自動
戻り値			
LONG	桁数		
解説			
桁数を設定する。			

プロパティ名	10. MdBcDecode_CheckITF										
概要	ITF チェック式を設定する。										
宣言形式	LONG GetMdBcDecode_CheckITF () void SetMdBcDecode_CheckITF (LONG inCheck)										
パラメータ	<table> <tr> <td>inCheck</td><td>チェック式</td><td>0x00000000</td><td>なし (既定値)</td></tr> <tr> <td></td><td></td><td>0x00000001</td><td>モジュラス 10 ウェイト 3</td></tr> </table>			inCheck	チェック式	0x00000000	なし (既定値)			0x00000001	モジュラス 10 ウェイト 3
inCheck	チェック式	0x00000000	なし (既定値)								
		0x00000001	モジュラス 10 ウェイト 3								
戻り値	LONG チェック式										
解説	チェック式を設定する。										

プロパティ名	11. MdBcDecode_CheckCODE39		
概要	CODE39 チェック式を設定する。		
宣言形式	<pre> LONG GetMdBcDecode_CheckCODE39 () void SetMdBcDecode_CheckCODE39 (LONG inCheck) </pre>		
パラメータ	inCheck	チェック式	0x00000000 なし (既定値) 0x00000040 モジュラス 43
戻り値	LONG	チェック式	
解説	チェック式を設定する。		

プロパティ名	12. MdBcDecode_CheckNW7		
概要			
NW-7 チェック式を設定する。			
宣言形式			
LONG GetMdBcDecode_CheckNW7 ()			
void SetMdBcDecode_CheckNW7 (LONG inCheck)			
パラメータ			
inCheck	チェック式	0x00000000	なし (既定値)
		0x00000001	モジュラス 10 ウェイト 3
		0x00000002	モジュラス 10 ウェイト 2
		0x00000004	ルーンズ
		0x00000008	モジュラス 11
		0x00000010	加重モジュラス 11
		0x00000020	モジュラス 16
		0x00000100	7 チェック DR
戻り値			
LONG	チェック式		
解説			
チェック式を設定する。			

プロパティ名	13. MdBcDecode_BinType		
概要			
2 値化種別を設定する。			
宣言形式			
LONG GetMdBcDecode_BinType() void SetMdBcDecode_BinType(LONG inType)			
パラメータ			
inType	2 値化種別	0x00000000	全体2値化（既定値） 画像全体から 2 値化閾値を決定します。 背景色が均一な画像に対して有効です。
		0x00000001	分割2値化 分割した局所領域から 2 値化閾値を決定します。 背景色にムラのある画像に対して有効です。
戻り値			
LONG	2 値化種別		
解説			
2 値化種別を設定する。			

プロパティ名	14. MdBcDecode_NoiseSize		
概要	ノイズ除去サイズを設定する。		
宣言形式	LONG GetMdBcDecode_NoiseSize() void SetMdBcDecode_NoiseSize(LONG inSize)		
パラメータ	inSize	ノイズサイズ	0～200 1/100mm 単位で指定。 縦＋横が指定サイズ以下のノイズ（黒画素）を無視する。 ノイズの混入による読取性能の低下を軽減します。 指定範囲：0～200（0：ノイズ除去なし） 初期値：30
戻り値	LONG	ノイズサイズ	
解説	ノイズ除去サイズを設定する。		

プロパティ名	15. MdBcDecode_HoleSize		
概要	穴埋めサイズを設定する。		
宣言形式	<pre> LONG GetMdBcDecode_HoleSize() void SetMdBcDecode_HoleSize(LONG inSize) </pre>		
パラメータ	inSize	穴サイズ	0～200 1/100mm 単位で指定。 縦＋横が指定サイズ以下の孔（白画素）を無視する。 かすれ等による読取性能の低下を軽減します。 指定範囲：0～200（0：穴埋めなし） 初期値：30
戻り値	LONG	孔サイズ	
解説	穴埋めサイズを設定する。		

プロパティ名	16. MdBcDecode_MinITF		
概要			
ITF 最小桁数を設定する。			
宣言形式			
LONG GetMdBcDecode_MinITF () void SetMdBcDecode_MinITF (LONG inMin)			
パラメータ			
inMin	ITF 最小桁数	2～30（偶数桁のみ）	指定の桁数に満たないコードは返しません。 初期値：4
戻り値			
LONG	桁数		
解説			
ITF 最小桁数を設定する。			

11. エラーコード一覧

エラーコード		
エラー番号		
ERROR_BC1_OK	0	// 正常終了
ERROR_BC1_ERROR	-100	// その他のエラー
ERROR_BC1_MEMORY	-101	// メモリエラー
ERROR_BC1_HANDLE	-102	// 無効ハンドル
ERROR_BC1_PARAM	-103	// 引数エラー (NULL ポインタ)
ERROR_BC1_IMAGE	-104	// 画像が不正
ERROR_BC1_BARTYPE	-105	// 種別エラー (非対応)
ERROR_BC1_DIGIT	-106	// 桁数処理エラー
ERROR_BC1_DECODE	-107	// デコード処理エラー
ERROR_BC1_CHECK	-108	// チェックエラー
// 共通エラーコード		
MDRC_OK	0	// 正常終了
MDRC_COMMON_ERR_BASE	-32000	// 共通エラーコード基準
MDRC_ERROR	-32000	// エラー
MDRC_MEMORY	-32001	// メモリ不足
MDRC_INVALIDHANDLE	-32002	// 無効ハンドル
MDRC_INVALIDPARAM	-32003	// パラメータエラー
MDRC_INVALIDPIC	-32004	// 無効画像データ
MDRC_FILEOPEN	-32005	// ファイルがオープンできない
MDRC_FILEREAD	-32006	// ファイル読み取りエラー
MDRC_FILEWRITE	-32007	// ファイル書き込みエラー
MDRC_BINPIC	-32008	// 2 値画像ではない
MDRC_CANNOT_LOAD_DLL	-32009	// DLL がロードできない
MDRC_NOTIMPL	-32010	// API が提供されていません
MDRC_ERR_LICENSE_PRODUCT	-32700	// ライセンスエラー・プロダクト
MDRC_ERR_LICENSE_DATE	-32701	// ライセンスエラー・有効期限エラー
MDRC_ERR_LICENSE_MACHINE	-32702	// ライセンスエラー・マシン ID
// カスタマバーコード エラーコード		
-32200	縦バーを検出するとき、輪郭画像を複製できません。	
-32201	縦バーを検出するとき、輪郭抽出に失敗しました。戻り値、ポインタ = %d, 0x%x	
-32202	傾いた矩形を傾かない矩形 (4 頂点) に変換するとき、左端 > 右端 です。	
-32203	傾いた矩形を傾かない矩形 (4 頂点) に変換するとき、上端 > 下端 です。	
-32204	矩形リストから DPI を求めるとき、グループ番号が不正です。グループ番号=%d	
-32205	矩形リストから DPI を求めるとき、矩形リストが空です。	
-32500	MdcCustomerBarcode_Find() のエラー規定番号	
-32501	MdcCustomerBarcode_Find() のメモリ	
-32502	MdcCustomerBarcode_Find() のその他のエラー	
-32503	MdcCustomerBarcode_Find() のハンドル	
-32504	MdcCustomerBarcode_Find() の文字列処理	
-32520	例外発生	
-32530	MdcCustomerBarcode_Open() のエラー規定番号	
-32531	MdcCustomerBarcode_Open() のメモリ	
-32532	MdcCustomerBarcode_Open() のその他のエラー	

-32533	MdcCustomerBarcode_Open () のハンドル
-32534	MdcCustomerBarcode_Open () の文字列処理
-32600	領域を検出するとき、解像度が不正です。
-32601	領域を検出するとき、領域数が不正です。
-32602	領域を検出するとき、方向が不正です。
-32603	領域を検出するとき、処理画像を保存できません。
-32604	領域を検出するとき、領域を検出できません。
-32605	認識するとき、領域を検出していません。
-32606	認識するとき、領域インデックスが不正です
-32607	認識するとき、文字配列が空です。
-32608	認識するとき、バーコード方向が不正です。
-32609	認識するとき、スペース数が%d 未満です。
-32610	認識するとき、スペース数が%d を超えました。
-32611	領域座標を取得するとき、領域を検出していません
-32612	領域座標を取得するとき、領域インデックスが不正です
-32613	領域座標を取得するとき、領域配列が空です。
-32614	処理画像の保存パス名を設定するとき、ロケールを設定できません。
-32615	処理画像の保存パス名を設定するとき、ファイル名の長さが不正です。
-32616	処理画像の保存パス名を設定するとき、パス名をシフト JIS に変換できません。
-32617	2 値画像を処理画像に変換するとき、画像の先頭アドレスが無効です。
-32618	2 値画像を処理画像に変換するとき、画像の幅が不正です。
-32619	2 値画像を処理画像に変換するとき、画像の高さが不正です。
-32620	2 値画像を処理画像に変換するとき、画像のバイト幅が不正です。
-32621	2 値画像を処理画像に変換するとき、解像度横が不正です。
-32622	2 値画像を処理画像に変換するとき、解像度縦が不正です。
-32623	2 値画像を処理画像に変換するとき、処理矩形の左端が不正です。
-32624	2 値画像を処理画像に変換するとき、処理矩形の上端が不正です。
-32625	2 値画像を処理画像に変換するとき、処理矩形の幅が不正です。
-32626	2 値画像を処理画像に変換するとき、処理矩形の高さが不正です。
-32627	2 値画像を処理画像に変換するとき、処理画像を作成できません。
-32628	2 値画像を処理画像に変換するとき、その他の例外が発生しました。
-32629	画像を-90 度回転するとき、回転画像を作成できません。
-32630	再領域検出するとき、点列を作成できません。
-32631	再領域検出するとき、輪郭画像を複製できません。
-32632	再領域検出するとき、輪郭抽出に失敗しました。
-32633	縦バーを検出するとき、輪郭画像を複製できません。
-32634	縦バーを検出するとき、輪郭抽出に失敗しました。
-32635	縦バーをグループ矩形に変換するとき、グループ別の点列を作成できません。
-32636	グループ矩形を結合するとき、処理画像を作成できません。
-32637	画素率を取得するとき、矩形の大きさが不正です。
-32638	画素率を取得するとき、点列を作成できません。
-32639	画素率を取得するとき、マスク画像を作成できません。
-32640	画素率を取得するとき、画素率 [0.0 - 1.0] が不正です。
-32641	画素率を取得するとき、メモリ例外が発生しました。
-32642	画素率を取得するとき、その他の例外が発生しました。
-32643	一致度を取得するとき、矩形数の重みが不正です。
-32644	四角形を傾かない矩形に変換するとき、整形画像を作成できません。
-32645	四角形を傾かない矩形に変換するとき、変換行列を作成できません。
-32646	画像の余白を削除するとき、輪郭画像を複製できません。
-32647	画像の余白を削除するとき、輪郭抽出に失敗しました。
-32648	画像の余白を削除するとき、マスク画像を作成できません。
-32649	画像の余白を削除するとき、左端が見付りません。
-32650	画像の余白を削除するとき、右端が見付りません。

-32651	画像の余白を削除するとき、左端が右端以上です。
-32652	画像の余白を削除するとき、上端が見付りません。
-32653	画像の余白を削除するとき、下端が見付りません。
-32654	画像の余白を削除するとき、上端が下端以上です。
-32655	画像の余白を削除するとき、削除画像を作成できません。
-32656	画像の余白を削除するとき、メモリ例外が発生しました。
-32657	画像の余白を削除するとき、その他の例外が発生しました。
-32658	画素密度を調整するとき、下限が上限より大きいです。
-32659	画素密度を調整するとき、最大処理回数が0以下です。
-32660	スペースを検出するとき、周辺分布メモリを確保できません。
-32661	スペースを検出するとき、平滑化メモリを確保できません。
-32662	スペースを検出するとき、平滑化に失敗しました。
-32663	スペースを検出するとき、2次平滑化メモリを確保できません。
-32664	スペースを検出するとき、2次平滑化に失敗しました。
-32665	スペースを検出するとき、差メモリを確保できません。
-32666	スペースを検出するとき、差の平滑化メモリを確保できません。
-32667	スペースを検出するとき、差の平滑化に失敗しました。
-32668	スペースを検出するとき、微分メモリを確保できません。
-32669	スペースを検出するとき、微分に失敗しました。
-32670	スペース数を調整するとき、入力スペース数が少な過ぎます。
-32671	スペース数を調整するとき、入力スペース数が多過ぎます。
-32672	スペース数を調整するとき、調整したスペース数が[%d]に一致しません。
-32673	バー種類を取得するとき、輪郭画像を複製できません。
-32674	バー種類を取得するとき、輪郭抽出に失敗しました。
-32675	バー種類を取得するとき、高さが不正です。
-32676	バー種類を取得するとき、メモリ例外が発生しました。
-32677	バー種類を取得するとき、その他の例外が発生しました。
-32678	k-means 法で分類するとき、分類数が不正です。
-32679	バー種類を3バーコードに変換するとき、スタート・ストップコードのロングバーが不正です。
-32680	バー種類を3バーコードに変換するとき(→)、開始・終了符号のセミロングバーが不正です。
-32681	バー種類を3バーコードに変換するとき(←)、開始・終了符号のセミロングバーが不正です。
-32682	バー種類を3バーコードに変換するとき(←→)、開始・終了符号のセミロングバーが不正です。
-32683	バー種類を3バーコードに変換するとき、バーコード方向が不正です。
-32684	バー種類を3バーコードに変換するとき、変換テーブルインデックスが不正です。
-32685	バー種類を3バーコードに変換するとき、[バー種類→コード]変換できません。
-32686	バー種類を3バーコードに変換するとき、チェックデジットが不正です。
-32687	3バーコードを文字コードに変換するとき、郵便番号が不正です。
-32688	3バーコードを文字コードに変換するとき、第二制御コードが不正です。
-32689	3バーコードを文字コードに変換するとき、英字が不正です。
-32690	3バーコードを文字コードに変換するとき、連続すべき空白コード(CC4)が不正です。

付録A. 認識サイズと解像度

認識可能なバーコードの最小の大きさ（目安）は、次の画像を A4 用紙に等倍率で印刷した場合、左から 300/200dpi でスキャンしたものとなる。

JAN13



ITF



CODE39



NW-7



GS1 DataBar Omnidirectional (RSS-14)



カスタマバーコード

（バーコードの端から端までが 64mm～88mm の範囲となるよう印刷してください）



<ご注意>

- 1.本書の内容の一部または全部を無断で転載することはお断りいたします。
- 2.本書に記載した内容は、将来予告なく変更することがあります。あらかじめご了承ください。
- 3.本書の内容について万一ご不審な点や誤りがありましたら、ご連絡下さるようお願いいたします
- 4.運用した結果の影響につきましては、責任を負いかねますのでご了承ください。

※ メディアドライブ株式会社の許可なく複製、改変などを行うことはできません。

第1版 2015年 3月
メディアドライブ株式会社