

Text Porter テキストポーター

V 5 . 3

導入ガイド

1.2 版

2013 年 03 月 25 日



改訂履歴

2013 年 03 月 11 日

V5.3 用の初版

2013 年 12 月 18 日

1.1 版

対応プラットフォームに、Windows 8.1(32bit/64bit)、
Windows Server 2012 R2(64bit)を追加。

2014 年 03 月 25 日

1.2 版

SPARC 版 Solaris 8, 9 を廃止。

目次

1 . パッケージに入っているもの.....	3
2 . 提供ファイルの構成	5
2.1 オブジェクトファイル.....	5
2.2 ヘッダファイル.....	7
2.3 文字コード変換用データ	7
2.4 サンプルソースコード.....	7
3 . モジュール詳細	9
4 . インストール.....	10
5 . 旧バージョンからの移行	12

1. パッケージに入っているもの

■ 製品構成

「TextPorterV5.3 サーバ版」について	1 枚
ソフトウェア使用許諾契約書兼顧客登録用紙	1 部
C D - R O M	1 枚

万一、不良品がありましたら直ちにお取り替えします。また不足のものがありましたらお送り致しますのでユーザサポート窓口までご連絡ください。

■ 製品内容

内容	説明
¥doc	マニュアルなどドキュメント一式。
¥Windows_X86_32	Windows X86 32bit 版ライブラリ、サンプルソースコード一式。
¥Windows_X86_64	Windows X86 64bit 版ライブラリ、サンプルソースコード一式。
¥Linux_X86_32	Linux X86 32bit 版ライブラリ、サンプルソースコード一式。
¥Linux_X86_64	Linux X86 64bit 版ライブラリ、サンプルソースコード一式。
¥Solaris_SP_32	Soalris SPARC 32bit 版ライブラリ、サンプルソースコード一式。
¥Solaris_SP_64	Soalris SPARC 64bit 版ライブラリ、サンプルソースコード一式。
¥Solaris_X86_64	Soalris X86 64bit 版ライブラリ、サンプルソースコード一式。

ライブラリとサンプルソースは、Windows 系は ZIP で、Unix 系は tar+gzip で圧縮されています。

■ 圧縮ファイルの展開

それぞれの圧縮ファイルをディレクトリ付きで展開すると、以下のディレクトリ(フォルダ)構成になります(以下の記述では、フォルダとディレクトリは同じ意味ですので、適宜読み替えてください)。

dmc_com	COM インターフェース ¹
dmc_conf	文字コード変換用定義データのパス取得関数
dmc_java	Java インターフェース ²
dmc_perl	Perl インターフェース ³
Include	ヘッダファイル
Lib	実行形式一式(コマンドライン、ライブラリ、定義データなど)
sample	サンプルアプリケーションソースコード

COM インターフェース、Java インターフェース、Perl インターフェースは、プラットフォームによってサポートしていないものもあります。

Windows 系の圧縮ファイル展開方法：

Windows 系の圧縮ファイルは、zip 圧縮してあります。

これを展開できるソフトは数多くありますので、それで展開してください。

Unix 系の圧縮ファイルの展開方法：

Unix 系の圧縮ファイルは、tar + gz 圧縮してあります。

これを展開できるソフトの一例は、gtar です(Linux では tar でも可能です)。

コマンド： `gtar -zxvf 圧縮ファイル名`

¹ Windows_X86_32 に入っています。

² Windows_X86_32, Windows_X86_64, Linux_X86_32, Linux_X86_64, Solaris_SP_32, Solaris_SP_64, Solaris_X86_64 に入っています。

³ Windows_X86_32, Linux_X86_32, Linux_X86_64, Solaris_SP_32, Solaris_SP_64 に入っています。

2. 提供ファイルの構成

2.1 オブジェクトファイル

*各プラットフォームフォルダの Lib フォルダに格納されております。

OS	CPU	Dynamic Link Library	Remarks
Windows	X86	識別子.dll	Windows XP(32/64bit) Windows Vista(32/64bit) Windows 7(32bit/64bit) Windows 8(32bit/64bit) Windows 8.1(32bit/64bit) Windows Server 2003(32/64bit) Windows Server 2003 R2(32/64bit) Windows Server 2008(32bit/64bit) Windows Server 2008 R2(64bit) Windows Server 2012(64bit) Windows Server 2012 R2(64bit) での動作保証 (動作には、msvcr90.dll, msvcp90.dllが必要) 本ライブラリは、Microsoft Visual C++ 2008 SP1 でビルドしています
Linux	X86	lib+識別子.so (シンボリックリンクが含まれます。)	GCC 4.1.2 以上での動作保証 (動作には、libc-2.5.so 以上, libstdc++.so.6 以上で、これらとバイナリ互換性があるライブラリが必要) 本ライブラリは、GCC 4.1.2 でビルドしています。
Solaris	SPARC	lib+識別子.so (シンボリックリンクが含まれます。)	Solaris10 以降 32bit/64bit での動作保証 (動作には、Sun C/C++ 5.8 以降のランタイムライブラリが必要) 本ライブラリは、Sun C/C++ 5.8 でビルドしています
	X86		Solaris10 64bit での動作保証 (動作には、Sun C/C++ 5.8 以降のランタイムライブラリが必要) 本ライブラリは、Sun C/C++ 5.8 でビルドしています

Windows 版の動作に必要なライブラリが、システムにインストールされていない場合は、製品パッケージの redistrib フォルダにある「Microsoft Visual C++ 2008 SP1 再頒布可能パッケージ」をインストールしてください。

32bit 版は vc_redist_x86.exe, 64bit 版は vc_redist_x64.exe です。

Linux 版の動作に必要なライブラリが、システムにインストールされていない場合、ディストリビュータから必要なものを取得し、インストールしてください。

動作保証については、対応プラットフォーム(OS, JavaVM など)に起因する問題は、弊社では保証できません。また、プラットフォームに起因する問題に対する解決先・回避策の提供は、

通常サポートには含まれません。

TextPorter の販売中、あるいは有償保守契約の期間中であっても、プラットフォーム製造元のサポート期間が終了した場合、動作保証はできません。

プラットフォーム製造元のサポート期間が終了したあとも、TextPorter の動作保証をお求めの場合は、弊社までご相談ください。

2.2 ヘッダファイル

インターフェース関数 / エラーコード等を記述したテキストファイル。

2.3 文字コード変換用データ

文字コード変換用データは外部ファイル^{*1}からテキスト抽出ライブラリの起動時に読み込みます。このため、テキスト抽出ライブラリの初期化時に外部ファイルのパスを取得する関数を用意しなければなりません。

参考のために、外部ファイルのパスを取得する関数のみが実装されたサンプルソースを添付しています。製品パッケージの dmc_conf フォルダを参照してください。

^{*1} 外部ファイルは、弊社より提供いたしました <¥Lib¥base2> フォルダ内に保存してあります。

関数仕様

```
int dmc_GetKeyValue( char *key, char *value, int nbyte );
```

【引 数】

- char *key: 検索するキー。定義データを保存したパスを取得する場合、キー値に "Charsetpath" を設定する。
- char *value: 検索された値を格納するバッファ。
- int nbyte: 検索された値を格納するバッファのサイズ

【戻り値】

- < 0: エラー
- > 0: 実際に value バッファに格納されたデータのサイズ。

key、value ともに ascii の null terminate string とします。

関数詳細

弊社は、文字コード変換用データの格納パスはカレントディレクトリ下の base2 という相対パスを仮定しています。

文字コード変換用データ (弊社より提供した ¥Lib¥base2 フォルダ下のもの) の格納パスが上記のデフォルトパスと異なる場合、dmc_conf のサンプルソースを参考に外部ファイルの保存先を実際の保存先に書き替え、ライブラリ dmc_conf を作成して、弊社提供のライブラリと差し替えていただくようお願いいたします。

文字コード変換用データの格納パスは大文字小文字の混在ができます。ファイル名は小文字のみサポートします。

2.4 サンプルソースコード

ライブラリの具体的な使用方法を記述したソースコードを提供します。

各プラットフォームフォルダ内の sample フォルダにあります。

サンプルは、無保証、無サポートです。また、予告なく、以前のバージョンとは非互換な

修正が行われる可能性もあります。あくまで、ライブラリ利用のサンプルである点、ご承知おきください。

注意：Windows_X86_64 版でストリームを使用する場合は、`setlocale()`にてロケールの設定を行ってください。

3. モジュール詳細

本ライブラリは以下の処理モジュールから構成されます。

- C インターフェース (C インターフェース提供)
- ライセンス管理モジュール (ご契約に応じて、使用可能な機能及び、使用期限などの情報を管理)
- ファイル識別モジュール (抽出元ファイルフォーマットの識別)
- テキスト抽出モジュール (各ファイルフォーマットからのテキスト抽出エンジン)
- 共通処理モジュール (文字列変換機能、共通利用する処理モジュール)
- 初期化用定義データパス取得用関数
- PDF 処理モジュール
- Office2007 処理モジュール
- ICU モジュール
- Java インターフェース (Java インターフェース提供)
- COM インターフェース (COM インターフェース提供)
- Perl インターフェース (Perl インターフェース提供)

モジュール名	ファイル名称
C インターフェース	dmc_txif
COM インターフェース	To_com
ライセンス管理モジュール	dmc_txli <u>ライセンス管理ファイル</u> dmc_txli.dat
ファイル識別モジュール	dmc_dtct
テキスト抽出モジュール	dmc_tx*
共通処理モジュール	dmc_comm/dmc_oscomm
初期化用定義データパス取得用関数	dmc_conf
PDF 処理モジュール	dmc_Pdfexploremp/dmc_PdfResmp /dmc_ahgralzwmp
Office2007 処理モジュール	OoxCommonTextPorter.dll XfoCommonTextPorter.dll
ICU モジュール	Icu****.dll
Java インターフェース	dmcjava.dll, dmcjava.jar
COM インターフェース	To_com.dll
Perl インターフェース	DMC_Perl.dll, DMC_Perl.pm

“*”にはアンテナハウスで定めるエンジン略称をモジュール毎に設定します。

COM インターフェースは、Windows 32bit のみ提供しています。

Java インターフェース、Perl インターフェースは、プラットフォームによってサポートしていないものもあります。詳細はお問い合わせください。

4. インストール

(1) Windows 系

*.dll をアプリケーションと同じフォルダにコピーしてください。

Windows のシステムフォルダ(例 WINDOWS や System32 など)にコピーするのは、トラブルの原因になりますので、避けてください。

格納したフォルダを、環境変数 PATH に指定して、dll が検索されるようにしてください。

文字コード変換用データの保存先フォルダのフルパスを、環境変数 DMC_TBLPATH で指定することができます。指定する際は、最後に ¥ をつけてください。指定しないと、ライブラリを格納したフォルダにある base2 を想定します。

COM インターフェースを使用する場合は、上記アプリケーションと同じフォルダに入れて、REGSVR32 で COM の登録を行ってください。

Java インターフェースを使用する場合、dmcjava.jar のあるフォルダと上記*.dll があるアプリケーションのフォルダが違えば、文字コード変換用データの保存先フォルダのフルパスを、環境変数 DMC_TBLPATH で指定してください。指定する際は、最後に ¥ をつけてください。

(2) Unix 系

*.so*をアプリケーションと同じディレクトリにコピーしてください。

Unix 系では、お客様がシンボリックリンクを設定しなくてもいいように、お客様の便宜を図るため、ライブラリをシンボリックリンク付きで提供しています。

libdmc_xxx.so.5.2	ライブラリの本体
libdmc_xxx.so.5	libdmc_xxx.so.5.2 へのシンボリックリンク
libdmc_xxx.so	libdmc_xxx.so.5 へのシンボリックリンク

といった具合です。

これを生かすためには、コピーの際には、シンボリックリンクをシンボリックリンクとしてコピーするオプションを使ってください。たとえば、Linux の cp コマンドでは、-d オプションを使えば、シンボリックリンクをシンボリックリンクとしてコピーできます。詳しくは、各プラットフォームのマニュアルを参照してください。

格納したディレクトリを、環境変数 LD_LIBRARY_PATH など、そのプラットフォームが共有ライブラリ検索に使用する環境変数に指定して、so が検索されるようにしておいてください。

文字コード変換用データの保存先ディレクトリのフルパスを、環境変数 DMC_TBLPATH で指定することができます。指定する際は、最後に / をつけてください。指定しないと、ライブラリを格納したディレクトリにある base2 を想定します。

Java インターフェースを使用する場合、dmcjava.jar のあるディレクトリと上記*.so*があるアプリケーションのディレクトリが違う場合は、文字コード変換用データの保存先ディレクトリのフルパスを、環境変数 DMC_TBLPATH で指定してください。指定する際は、最後に / をつけてください。

(3) ライセンス管理ファイル dmc_txli.dat

- a) dmc_txli.dat はアプリケーションと同じフォルダに入れることができます。
- b) dmc_txli.dat は文字コード変換用データの保存先と同じフォルダに入れることができます。

本ライブラリは、アプリケーション格納フォルダ 文字コード変換用データの保存フォルダ順で dmc_txli.dat を検索します。

dmc_txli.dat をアプリケーションと同じフォルダに入れる場合、Windows 上で、プログラム実行中にカレントディレクトリがアプリケーション格納フォルダから他のフォルダに変更される可能性があります。そのため、dmc_txli.dat が見つからなくなってしまう、2002 というエラーが発生します。そのような場合は、dmc_txli.dat を文字コード変換用データの保存先に入れて、DMC_TBLPATH を設定した上で試してください。

5. 旧バージョンからの移行

V4 からの移行

V5 は、コンパイラの変更を行いましたので、V4 とはバイナリ互換性がありません。

Lib フォルダに格納されているモジュールを差し替えるだけでは使用できません。

クリーンな再ビルドが必要です。本ライブラリのビルド環境については、「2.1 オブジェクトファイル」を参照してください。

API は、V4 用の API が用意されていますが、V5 の API に移行することを強く推奨します。V4 からの変更点については、V5-changes-from-V4_2.pdf を参照してください。