

## Perl インターフェイス利用ガイド

### ディレクトリ/ファイル構成

#### Windows\_X86\_32

DMC_Perl.dll	DMC_Perl の DLL
DMC_Perl.pm	DMC_Perl モジュールファイル
DMC_Perl_wrap.c	DMC_Perl インタフェイスラッパーのソース
Makefile.PL	Makefile 作成用のスクリプト
MANIFEST	Makefile.PL で使うマニフェスト
README	Makefile.PL で使う README
SampleScript¥	サンプルスクリプトが入ったディレクトリ
text_oem.h	DMC ライブラリのヘッダファイル

#### Windows\_X86\_32 以外

DMC_Perl.pm	DMC_Perl モジュールファイル
DMC_Perl_wrap.c	DMC_Perl インタフェイスラッパーのソース
Makefile.PL	Makefile 作成用のスクリプト
MANIFEST	Makefile.PL で使うマニフェスト
README	Makefile.PL で使う README
SampleScript¥	サンプルスクリプトが入ったディレクトリ
text_oem.h	DMC ライブラリのヘッダファイル

### 動作環境

Perl インターフェイスを利用するためには、DMC ライブラリが正常にインストールされ、利用できるようなっている必要があります。

Perl については、Windows であれば ActivePerl、それ以外では、Perl(そのプラットフォーム用の ActivePerl でも可)がインストールされている必要があります。64bit プラットフォームで利用するには、64bit 版の Perl が必要です。

## インストール

- Windows\_X86\_32

### 通常のインストール

動作環境に DMC\_Perl をインストールする場合は、Microsoft Visual C++(以下、VC) がインストールされている必要があります。

インストールには、VC に含まれる nmake をコマンドラインで使用するため、コマンドプロンプト起動後に、VC のインストールされたディレクトリの中にある VCVARS32.BAT を実行しておいて下さい。

```
dmc_perl において、  
copy ..¥Lib¥dmc_txif* .  
perl Makefile.PL
```

```
nmake  
nmake install
```

を実行することで、モジュールのビルドおよびインストールを行うことができます。

### 手動インストール

VC が無い等の理由で、手動でインストールを行う場合は、ActivePerl のインストールされたディレクトリが、<parldir>とした場合、<perldir>¥site¥lib¥DMC\_Perl¥に、DMC\_Perl.pm を、<perldir>¥site¥lib¥auto¥DMC\_Perl¥に DMC\_Perl.dll をコピーしてください。

ただし、DMC\_Perl.dll には、perl?.lib(??部は ActivePerl のバージョンによって異なる)をリンクしているので、動作環境にインストールされている ActivePerl 用の perl?.lib がリンクされた DMC\_Perl.dll を用意する必要があります。

### 動作確認を行った ActivePerl のバージョン

ActivePerl 5.8.8

- Windows\_X86\_32 以外  
dmc\_perl ディレクトリにおいて、  
cp ../Lib/libdmc\_txif\* .  
perl Makefile.PL  
make  
make install  
を実行することで、モジュールのビルドおよびインストールを行うことができます。

**\* make install 時は、root 権限で行ってください。**

インストール時の make でインストール環境に併せてコンパイルを行うので、  
利用する Perl をビルドした C コンパイラーが必要となります。

#### **動作確認を行った Perl のバージョン**

Perl5.8 や ActivePerl 5.8.8

## インターフェイス

Perl 用のインターフェイスは、DMC ライブラリの公開 API の各関数に対応しています。各インターフェイス関数には、サンプルスクリプトを用意しているので、そちらも合わせて参照してください。

### 共通事項

DMC\_Perl を使用するためには、スクリプトに

```
use DMC_Perl::DMC_Perl;  
を記述しておいてください。
```

### 制限事項

Perl インターフェイスは、ストリームへの出力に対応していません。

## 構造体

構造体を使用する場合は、事前にインスタンスを取得する必要があります。また、取得したインスタンスは必要なくなったら解放しなければなりません。

例) DMC\_TEXTINFO 構造体の場合

```
Stinfo = new DMC_Perl::DMC_TEXTINFO();           # インスタンスの取得  
Stinfo->{GroupName} = "Shift_JIS";              # 構造体メンバへの代入  
                                                  # Byte *メンバの場合は、この記  
                                                  # 述でバッファの確保も行われます  
.  
.  
.  
undef $stinfo->{GroupName};                       # バッファの解放  
Stinfo->DESTROY();                                # インスタンスの解放定数
```

text\_oem.h に定義されている定数を使用する場合は、“\$DMC\_Perl::定数名”と記述します。

int \*

DMC\_GetPageText\_V4()等に引数として渡す必要がある int \*は、DMC\_Perl::intp()関数を使ってインスタンスを取得します。

例)

```
$pages = new DMC_Perl::intp(); # インスタンスの取得  
$pages->assign(5);            # 変数へ値の代入(C では、  
                              *pages = 5 に相当する)
```

```

print $pages->value();           # 値の取得
$pages->DESTROY();              # インスタンスの解放
コールバック関数

```

DMC\_GetPageText\_V4 等の最後の引数に渡すコールバック関数は、「\$DMC\_Perl::CallBackFunc」を固定で渡すこと。これ以外を渡す場合には、動作の保証はされない。また、この関数の処理は、addcbfunc.pl を参照のこと。処理内容を書き換えたい場合は addcbfunc.pl を書き換えて、VC でリビルドし DMC\_Perl\_wrap.c を生成することにより可能である。

### 関数一覧

関数名	サンプルスクリプトファイル名
DMC_Perl::DMC_GetFileInfo_V4	GetFileInfov4.pl
DMC_Perl::DMC_GetText_V4	GetTextv4.pl
DMC_Perl::DMC_GetProperty_V4	GetPropertyv4.pl
DMC_Perl::DMC_GetPageText_V4	GetPageTextv4.pl
DMC_Perl::DMC_GetPwdText_V4	GetPwdTextv4.pl
DMC_Perl::DMC_GetPwdPageText_V4	GetPwdPageTextv4.pl
DMC_Perl::DMC_GetPwdProperty_V4	GetPwdProperty.v4pl

例)テキスト抽出を行う場合

SampleScript フォルダにて、"Perl -Tw GetTextv4.pl 抽出元ファイル 抽出先ファイル"