

拡張モジュールのインストール

マルチバイト関数

①Red Hat Linux環境

第2章で登場したマルチバイト関数は拡張関数です。この関数を使用するためには、PHPのconfigure時に`--enable-mbstring`オプションを指定するだけでよく、大抵の環境ではこのオプションをつけていますので、他の拡張関数と異なり、あらためてインストールする必要はありません。

②Windows環境

第2章で登場したマルチバイト関数は拡張関数です。

「`%PHP_HOME%/extensions`(`%PHP_HOME%`はPHPを展開したフォルダとします)」内に`php_mbstring.dll`が存在することを確認した上で、`php.ini`内に以下の記述を追記して、DLL呼び出しを有効にしてください。

```
extension=php_mbstring.dll
```

その際、Apacheサーバを再起動することを忘れないようにしてください。

ZLib圧縮関数

①Red Hat Linux環境

第4章で登場したZLib圧縮関数は、拡張関数です。この関数を使用するためには、PHPのconfigure時に`--with-zlib`オプションを指定するだけでよく、大抵の環境ではこのオプションをつけていますので、他の拡張関数と異なり、あらためてインストールする必要はありません。

②Windows環境

第4章で登場したzLib圧縮関数は、PHP 4.2.X以前には拡張関数でしたが、PHP 4.3.XでPHPバイナリに組み込まれています。`php.ini`であらためてDLL呼び出しを有効にする必要はありませんので、注意してください(`php.ini`で`extension`パラメータを指定した場合、Apacheの再起動時にエラーが発生します)。

現在の環境がPHP 4.2.X以前である場合にのみ、「`%PHP_HOME%/extensions`(`%PHP_HOME%`はPHPを展開したフォルダとします)」内に`php_zlib.dll`が存在することを確認した上で、`php.ini`内に以下の記述を追記して、DLL呼び出しを有効にする必要があります。

```
extension=php_zlib.dll
```

その際、Apacheサーバを再起動することを忘れないようにしてください。

メール関数

①Red Hat Linux環境

Red Hat Linux版のPHPのメール関数は、内部のsendmailコマンドを用いてメールを送信します。

sendmailは特に指定しない限り、Red Hat Linux 9.0をインストールした時にデフォルトでインストールされます。

サンプル使用に際しては、php.iniのsendmail_pathパラメータをあらかじめ設定しておく必要があります。たとえば、以下のように設定します。

```
sendmail_path="/usr/sbin/sendmail -t -i -fhoge@hoge.com"
```

-tオプションは宛先をTO:ヘッダから読み込むことを、-iオプションは標準入力からメッセージを読み込む場合に「.」だけの行を入力を終端として扱わないことを、それぞれ意味します。
-fオプションは送付元のメールアドレスを指定します。「-f」とメールアドレスの間には、スペースが入らない点に注意してください。

なお、php.iniを新たに更新した場合には、Apacheサーバの再起動が必要になります。

②Windows環境

Windows版のPHPのmail関数は、外部のSMTPサーバを用いてメールを送信します。

php.iniのSMTP・sendmail_fromパラメータをあらかじめ設定しておく必要があります。たとえば、以下のように設定します。

```
SMTP=smtp.jp.yama.com  
sendmail_from=CQW15204@nifty.com
```

php.iniを新たに更新した場合には、Apacheサーバの再起動が必要になりますので、注意してください。

①Red Hat Linux環境

第5章で登場したClient URL Library関数は、拡張関数です。使用に際しては、事前に拡張モジュールをインストールしておく必要があります。最新のソースコード (curl-7.10.5.tar.gz) は、以下のURLから入手することができます。

```
http://curl.haxx.se/download.html
```

①cURLモジュールのインストールを行う

```
$ tar zxvf curl-7.10.5.tar.gz
$ cd curl-7.10.5
$ ./configure
$ make
$ su
# make install
```

②PHP用に拡張モジュール化する

```
$ cd PHP_TOP_DIR/ext/curl
$ phpize
$ ./configure
$ make
$ su
# make EXTENSION_DIR=/usr/local/lib/php/extensions install
```

PHP_TOP_DIRは、PHPが解凍されたディレクトリに応じて変更します。これにより、動的モジュールcurl.soが「/usr/local/lib/php/extensions」にインストールされます。

③php.iniのextensionパラメータを追加する

```
extension= curl.so
```

その際、Apacheサーバを再起動することを忘れないようにしてください。

②Windows環境

第5章で登場したClient URL Library関数は、拡張モジュールです。

「%PHP_HOME%/extensions(%PHP_HOME%はPHPを展開したフォルダとします)」フォルダ内にphp_curl.dllが存在することを確認した上で、php.ini内に以下の記述を追記して、DLL呼び出しを有効にする必要があります。

```
extension=php_curl.dll
```

その際、Apacheサーバを再起動することを忘れないようにしてください。

IMAP関数

①Red Hat Linux環境

第5章で登場したIMAP関数は、拡張関数です。使用に際しては、事前に拡張モジュールをインストールしておく必要があります。

最新のソースコード (imap-2001a.tar.Z)は、以下のURLから入手することができます。

```
ftp://ftp.cac.washington.edu/imap/
```

①IMAPモジュールのインストールを行う

```
$ tar zxvf imap-2001a.tar.Z
$ cd imap-2001a
$ make slx
$ su
$ cp c-client/c-client.a /usr/local/lib/
$ cp c-client/rfc822.h c-client/mail.h c-client/linkage.h
c-client/imap4rl.h /usr/local/include/
$ cd /usr/local/lib
$ ln -s c-client.a libc-client.a
$ /sbin/ldconfig
```

makeコマンドの後のオプション(slx)は、自分の環境に合わせて変更する必要があります。詳しくは、Makefileを参照してください('slx'は、glibc Linux and/or shadow password securityのあるLinuxに用います)。

なお、IMAPモジュールにはmake installがないので、手動でインストールを行う必要があります。

②PHP用に拡張モジュール化する

```
$ cd PHP_TOP_DIR/ext/imap
$ phpize
$ ./configure
$ make
$ su
# make EXTENSION_DIR=/usr/local/lib/php/extensions install
```

PHP_TOP_DIRは、PHPが解凍されたディレクトリに応じて変更します。これにより動的モジュールimap.soが「/usr/local/lib/php/extensions」にインストールされます。なお、make作業がうまくいかない場合には、config_vars.mkのINCLUDESにimapソース中のインクルードファイルが格納されているパスを追加した上で、再度makeを実行してください。

③php.iniのextensionパラメータを追加する

```
extension= imap.so
```

その際、Apacheサーバを再起動することを忘れないようにしてください。

②Windows環境

第5章で登場したIMAP関数は、拡張関数です。「%PHP_HOME%/extensions(%PHP_HOME%はPHPを展開したフォルダとします)」内にphp_imap.dllが存在することを確認した上で、php.ini内に以下の記述を追記して、DLL呼び出しを有効にする必要があります。

```
extension=php_imap.dll
```

その際、Apacheサーバを再起動することを忘れないようにしてください。

PDF関数

PDFLibはシェアウェアです。実運用においては、PDFLib社 (<http://www.pdflib.com/pdflib/index.html>) に対してライセンス料が発生しますので、注意してください

①Red Hat Linux環境

第7章で登場したPDF関数は、拡張関数です。使用に際しては、事前に拡張モジュールPDFLib+PDI+libjpeg+libtiffをインストールしておく必要があります(ただし、libjpeg+libtiffは、Linux上にデフォルトでインストールされています)。

最新のソースコードは、以下のURLから入手することができます。

```
http://www.pdflib.com/products/pdflib/download/      (PDFLib)
ftp://ftp.uu.net/graphics/jpeg/                      (libjpeg)
ftp://ftp.remotesensing.org/pub/libtiff/              (libtiff)
```

①libjpegモジュールのインストールを行う

```
$ tar zxvf jpegsrc.v6b.tar.gz
$ cd jpeg-6b
$ ./configure
$ make
$ su
# make install
```

②libtiffモジュールのインストールを行う

```
$ tar zxvf tiff-v3.5.7.tar.gz
$ cd tiff-v3.5.7
$ ./configure
$ make
$ su
# make install
```

③PDFLib+PDI(PDF Import library)モジュールのインストールを行う

PDFLib+PDIモジュールについては、配布されているパッケージにすでにコンパイル済みのバイナリファイルが含まれています。そこで、入手したダウンロードファイルPDFLib-5.0.1-Linux.tar.gzを解凍し、配下の「/PDFLib-5.0.1-Linux/bind/php/php-4.2.1」

ディレクトリ配下に含まれる「libpdf_php.so」を「/usr/local/lib/php/extensions」ディレクトリにコピーします。

④php.iniのextensionパラメータを追加する

```
extension=libpdf_php.so
```

その際、Apacheサーバを再起動することを忘れないようにしてください。

②Windows環境

第7章で登場したPDF関数は、拡張関数です。使用に際しては、事前に拡張モジュールPDFLib+PDIをインストールしておく必要があります。

最新のソースコードは、以下のURLから入手することができます。

```
http://www.pdflib.com/products/pdflib/download/
```

①ダウンロードしたモジュールPDFlib-5.0.1-Windows.zipを適当なフォルダに格納する

「PDFlib-5.0.1-Windows」フォルダ配下に各種ファイルが展開されます。

その中の「bind/php/php421」フォルダ内のlibpdf_php.dllを、「%PHP_HOME%/extensions(%PHP_HOME%はPHPを展開したフォルダとします)」内にコピーします。

②php.iniを編集する

php.ini内に以下の記述を追記して、DLL呼び出しを有効にします。

```
extension=libpdf_php.dll
```

その際、Apacheサーバを再起動することを忘れないようにしてください。

①Red Hat Linux環境

第7章で登場したNamazu関数は、拡張関数です。使用に際しては、全文検索システムNamazu、漢字 かな変換プログラムKAKASI、およびCVS上で公開されている拡張モジュールを事前にインストールしておく必要があります。

最新のソースコードは、以下のURLから入手することができます。

```
http://www.namazu.org/                (namazu-2.0.12)
http://kakasi.namazu.org/              (kakasi-2.3.4)
http://chora.php.net/cvs.php/pecl/namazu
                                         (拡張モジュールのソースコード式)
```

①Namazu・Kakasiのインストール状況を確認する

```
$ rpm -qa | grep namazu
$ rpm -qa | grep kakasi
```

環境にインストールされていないことを確認した後、上記のURLから入手したソースコードを用いてインストールを行います。

なお、日本語環境でNamazuをインストールするには、あらかじめ以下のソフトウェアをインストールしておく必要があります。

- ・ Perl(5.004以降)
- ・ File::MMagic
- ・ nk(1.71を推奨)

これらは、環境に合わせて足りないものを、必要に応じてインストールしてください。

②Kakasiのインストールを行う

上記のURLよりKakasiに関するソースコードを入手して、インストールを行います。

```
$ tar zxvf kakasi-2.3.4.tar.gz
$ cd kakasi-2.3.4
$ ./configure
$ make
$ su
# make install
```


③Namazuのインストールを行う

P.470のURLよりNamazuに関するソースコードを入手して、インストールを行います。

```
$ tar zxvf namazu-2.0.12.tar.gz
$ cd namazu-2.0.12
$ ./configure
$ make
$ su
# make install
```

④Namazuの環境設定を行う

Namazuの設定ファイルであるnamazurcを書き換えます。

```
$ cd /usr/local/etc/namazu/
$ cp namazurc-sample namazurc
```

テキストエディタなどでnamazurcを開き、

```
#Index          /usr/local/var/namazu/index
#Replace        /home/foo/public_html/  http://www.foo.bar.jp/~foo/
#Lang           ja
#EmphasisTags   "<strong class=¥"keyword¥">"   "</strong>"
```

の#を削除します。また、ふたつ目の

```
Replace        /home/foo/public_html/  http://www.foo.bar.jp/~foo/
```

は

```
Replace        /usr/local/apache2/htdocs/ /
```

と書き換えます。

⑤全文検索用インデックスを作成する

Namazuの設定ファイルnamazurcに従い、Indexの場所を

```
/usr/local/var/namazu/index
```

とします(サンプル中のnmz_open関数についても、同様に書き換えてください)。

```
$ mkdir -p /usr/local/var/namazu/index
$ mknmz -O /usr/local/var/namazu/index /usr/local/apache2/
htdocs/petit/chap7/doc
```



⑥ NamazuをPHPの拡張モジュールとして組み込む

P.470のサイトからダウンロードしたソースコード一式を、任意のディレクトリ内に移動します(以下では展開先を「namazu」とします)。

```
$ cd namazu
$ phpize
$ ./configure --with-namazu
$ make
$ su
# make EXTENSION_DIR=/usr/local/lib/php/extensions install
```

PHP_TOP_DIRは、PHPが解凍されたディレクトリに応じて変更します。これにより動的モジュールnamazu.soが「/usr/local/lib/php/extensions」にインストールされます。

⑦ php.iniのextensionパラメータを追加する

```
extension=namazu.so
```

その際、Apacheサーバを再起動することを忘れないようにしてください。

② Windows環境

Namazuを使用するには、以下のプログラムをあらかじめ入手して、インストールしておく必要があります。最新のソースコードは、以下のURLから入手することができます。

① Active Perlをインストールする

Namazuの環境設定のためにPerlインタプリタが必要になります。最新のインストールファイルActivePerl-5.6.1.633-MSWin32-x86.msiは、以下のURLから入手することが可能です。

```
http://www.ActiveState.com/ActivePerl/
```

ダウンロードしたファイルは、ダブルクリックするだけでインストールが実行されます。途中で聞かれる設定値は、基本的にすべてデフォルトのままでOKです。

② Namazu for Win32をインストールする

Namazu本体です。最新のファイルは、以下のURLから入手可能です。

```
http://www.namazu.org/windows/
```

ダウンロードしたファイルは自己解凍ファイルですので、ダブルクリックするだけでインストールが完了します。必要な環境変数なども自動的に書き込まれますので、ほとんど気にする必要はありません。

コマンドプロンプトから

```
> namazu --help
```

を実行した時に、コマンドの説明が表示されれば、インストールは成功です。

③ Kakashi for Win32をインストールする

Namazuのインデックス作成のための辞書ファイルとして必要になります。最新のファイルは、以下のURLから入手することが可能です。

```
ftp://kakasi.namazu.org/pub/kakasi/win32/
```

ダウンロードしたファイル(kakasi-2.3.4.zip)は、「C:\kakashi」に展開します(他のフォルダに展開することも可能ですが、その場合、一部の環境変数を書き換える必要が出てまいりますので、あまりお勧めできません)。

④ インデックスを作成する

すべてのソフトがインストールできたら、次はいよいよインデックスの作成です。

たとえば、「C:\Program Files\Apache Group\Apache2\htdocs\sample\chap3\namazu.doc」配下にある文書に対して全文検索を行いたい場合、コマンドプロンプトから以下のようなコマンドを実行します。

```
> mknmz -O C:\namazu\var\namazu\index "C:\Program Files\Apache Group\Apache2\htdocs\petit\chap7.doc"
```

これによって、「C:\Program Files\Apache Group\Apache2\htdocs\petit\chap7.doc」配下のインデックスが「C:\namazu\var\namazu\index」に作成されます。「C:\namazu\var\namazu\index」は、そのままnmz_open関数で使用する引数になります。

コマンドプロンプトから以下のコマンドを実行した時、「ASP」という言葉が含まれる文

書の一覧が表示されれば、インデックスの作成は成功です。

```
namazu "ASP"
```

⑤ 拡張モジュールを有効にする

最後に、PHP用のNamazu拡張モジュールを設定しておきましょう。最新のファイルは、以下のURLからダウンロード可能です。

```
http://tag3.hp.infoseek.co.jp/
```

解凍したファイルの中身を「%PHP_HOME%/extensions\%PHP_HOME%はPHPを展開したフォルダとします」内にコピーして、php.ini内に以下の記述を追記します。

```
extension=php_namazu.dll
```

その際、Apacheサーバを再起動することを忘れないようにしてください。

イメージ関数

① Red Hat Linux版

第7章で登場したイメージ関数は、拡張関数です。使用に際しては、事前に拡張モジュールGDが必要となります。また、GD画像関数を使うには、関連するライブラリ(zlib・libpng・libjpeg)をインストールする必要があります。

最新のソースコードは、以下のURLから入手することができます。

```
ftp://ftp.uu.net/graphics/jpeg/ (libjpeg)
http://www.libpng.org/pub/png/libpng.html (libpng)
```

本書では、ZLibライブラリはPHPのインストール時に静的に組み込んでいますので、あらためてインストールする必要はありません。また、GDもPHP組み込みのものを使用することにします(PHP4.3.X以降)

① libjpegをインストールする

```
$ tar zxvf jpegsrc.v6b.tar.gz
$ cd jpeg-6b
$ ./configure --enable-shared --enable-static
$ make
```

```
$ su
# make install
```

②libpngをインストールする

```
$ tar zxvf libpng-1.2.5.tar.gz
$ cd libpng-1.2.5
$ ln -s scripts/makefile.linux MakeFile
$ su
# make install
```

libpngではインストールする環境や、配置するライブラリの種類などによって、MakeFileが分かれて存在します。自分の環境に合わせて変更してください。ここでは、Linux用でlibpng.α(static library)・libpng.so.2.1.0.12(shared library)を生成するMakeFileを用いています。

③PHP本体に組み込む

PHP本体のconfigure時にGDの共有モジュールを生成する次のオプションを追加して、インストールを行います。

```
$ ./configure --enable-trans-sid
               --with-apxs2=/usr/local/apache2/bin/apxs
               --enable-mbstring
               --enable-mbstr-enc-trans
               --with-mysql=/usr/local/mysql
               --with-zlib
               --with-gd
               --with-jpeg-dir
               --with-png-dir
```

以降の設定は、P.11の環境設定の手順に従ってください。

②Windows環境

第7章で登場したイメージ関数は、拡張モジュールです。「%PHP_HOME%/extensions(%PHP_HOME%はPHPを展開したフォルダとします)」内にphp_gd2.dllが存在することを確認した上で、php.ini内に以下の記述を追記して、DLL呼び出しを有効にする必要があります。

```
extension=php_gd2.dll
```

その際、Apacheサーバを再起動することを忘れないようにしてください。

プリンタ関数

①Windows環境

第7章で登場するプリンタ関数は、拡張モジュールです。

php.ini内でコメントアウトされている以下の記述から、

```
extension=php_printer.dll
```

先頭の「;」を削除して、有効にする必要があります。

その際、Apacheサーバを再起動することを忘れないようにしてください。

DOM XML関数

①Red Hat Linux環境

第7章で登場したDOM XML関数は、拡張関数です。使用に際しては、事前に拡張モジュールをインストールしておく必要があります。

最新のソースコードは、以下のURLから入手することができます。

```
ftp://xmlsoft.org/
```

①domxmlモジュールのインストールを行う

```
$ tar zxvf libxml2-2.5.7.tar.gz
$ cd libxml2-2.5.7
$ ./configure
$ make
$ su
# make install
```

②PHP用に拡張モジュール化する

```
$ cd PHP_TOP_DIR/ext/domxml
$ phpize
```

```
$ ./configure --with-zlib-dir
$ make
$ su
# make EXTENSION_DIR=/usr/local/lib/php/extensions install
```

PHP_TOP_DIRはPHPがインストールされたディレクトリに応じて変更します。これにより動的モジュールdomxml.soが「/usr/local/lib/php/extensions」にインストールされます。

③php.iniのextensionパラメータを追加する

```
extension= domxml.so
```

その際、Apacheサーバを再起動することを忘れないようにしてください。

②Windows環境

第7章で登場したDOM XML関数は、拡張モジュールです。「%PHP_HOME%/extensions (%PHP_HOME%はPHPを展開したフォルダとします)」内にphp_domxml.dllが存在することを確認した上で、php.ini内に以下の記述を追記して、DLL呼び出しを有効にする必要があります。

```
extension=php_domxml.dll
```

その際、Apacheサーバを再起動することを忘れないようにしてください。

XSLT関数

①Red Hat Linux環境

第7章で登場したXSLT関数は、拡張関数です。使用に際しては、事前に拡張モジュールlibiconv・expat、およびSabltronをインストールしておく必要があります。最新のソースコードは、以下のURLから入手することができます。

http://www.gnu.org/software/libiconv/	(libiconv)
http://sourceforge.net/projects/expat/	(expat)
http://www.gingerall.com/	(Sabltron)

① libiconvをインストールする

```
$ tar zxvf libiconv-1.8.tar.gz
$ cd libiconv-1.8
$ ./configure
$ make
$ su
# make install
```

② expatをインストールする

```
$ tar zxvf expat-1.95.5.tar.gz
$ cd expat-1.95.5
$ ./configure
$ make
$ su
# make install
```

③ Sablotronをインストールする

```
$ tar zxvf Sablot-0.98.tar.gz
$ cd Sablot-0.98
$ ./configure
$ make
$ su
# make install
```

④ PHP用に拡張モジュール化する

```
$ cd PHP_TOP_DIR/ext/xslt
$ phpize
$ ./configure --enable-xslt --with-xslt-sablot --with-
expat-dir=/usr/local
$ make
$ su
# make EXTENSION_DIR=/usr/local/lib/php/extensions install
```



PHP_TOP_DIRは、PHPがインストールされたディレクトリに応じて変更します。これにより動的モジュールxslt.soが「/usr/local/lib/php/extensions」にインストールされます。

⑤php.iniのextensionパラメータを追加する

```
extension= xslt.so
```

その際、Apacheサーバを再起動することを忘れないようにしてください。

②Windows環境

本項で用いているXSLT関数は、拡張モジュールです。「%PHP_HOME%/extensions (%PHP_HOME%はPHPを展開したフォルダとします)」内にphp_xslt.dllが存在することを確認した上で、php.ini内に以下の記述を追記して、DLL呼び出しを有効にする必要があります。

```
extension=php_xslt.dll
```

その際、Apacheサーバを再起動することを忘れないようにしてください。