

PHP マニュアル

インストール環境構築 for Windows, Linux

PHP のインストールから環境設定まで

- 1.PHP とはなにか？
- 2.PHP の仕様
- 3.Apache のインストール
- 4.PHP のインストール
- 5.PHP の環境設定 (PHP.ini の設定)
- 6.Apache の設定 (httpd.conf の設定)
- 7.PHP 動作確認
- 8.Linux 環境下での LAMP 構築
 - ・Apache のインストール状況の確認
 - ・PHP のインストールと設定
 - ・PHP のインストール
 - ・php.ini の設定
 - ・httpd.conf の設定
 - ・動作確認

1.PHP とはなにか？

PHP とは、動的に HTML データを生成することによって動的なウェブページ(同じウェブページでもアクセスするたびに表示が更新されているウェブページのこと)を実現することを主な目的としたプログラミング言語です。Web ページの開発をメインとした利用を想定して作られた言語ですが、バッチ処理(大量のデータを一度に処理するためのプログラム)などの処理も作成可能です。今回は、Web ページの開発である HTML 埋め込み型をメインに話を進めていこうと思います。PHP で作成できる代表的なアプリケーションとしては電子掲示板システム、アクセスカウンタ、ブログシステムなどがあります。規模としては小～中規模のシステム開発に利用される言語です。PHP のバージョン5からオブジェクト指向というアルゴリズム(考え方)が強化されたため、Java などの大規模システム開発用の言語と遜色のないレベルまで成熟してきました。今後、より開発の現場で使用される機会が多くなると期待されています。

2.PHP の仕様

PHP は Common Gateway Interface(コモン・ゲートウェイ・インターフェース、CGI)と呼ばれるウェブサーバ上でプログラムを動作させるための仕組みを採用しています。このような言語をサーバサイドプログラミング言語といいます。具体的に Web ブラウザからの処理要求を受け、その処理をウェブサーバ側で処理をし結果を HTML 形式へと変換したのちにそれを要求もとの Web ブラウザへと返します。サーバサイドプログラミング言語の利点としてはサーバ側で処理をするのでさまざまな異なる種類の Web ブラウザから処理要求を出しても統一された処理結果が返ってくる点です。これに対し、JavaScript などのクライアントサイドプログラミング言語ではブラウザで処理を行うためブラウザのバージョンが古かったり開発時に想定していたブラウザと異なるものを使っていると出力時のレイアウトが崩れたり、最悪の場合動作しないなどの問題が起こります。

近年では、Web サーバのプロセスとしてインタプリタを常駐させておくことにより、CGI からプログラムを呼び出すオーバーヘッドを減らし、パフォーマンスを向上させた Java Servlet や mod_perl、mod_php、FastCGI などの実装も出現しており従来よりも高速に処理を完了させることができます。

また、処理ロジック、レイアウトの出力、ユーザからの入力といったいわゆる MVC モデル(モデル・ビュー・コントローラ)を使用することができるテンプレートエンジンの Smarty などをはじめ早く正確に、また見やすいコードを記述でき保守に役立つフレームワークと呼ばれるものが PHP では利用可能になっています。

3.Apache のインストール

PHP をインストールするにはまず、Web サーバー用ソフトウェアをインストールする必要があります。今回は代表的な Web サーバー用ソフトウェアとして Apache(アパッチ)をインストールしてみます。
















➤ Windows 用パッケージの取得

Apache を取得するため以下のサイトにアクセスします。

<http://www.apache.jp/misc/download.html>

次にリンク先をたどり binaries フォルダ以下の win32 フォルダを開きます。

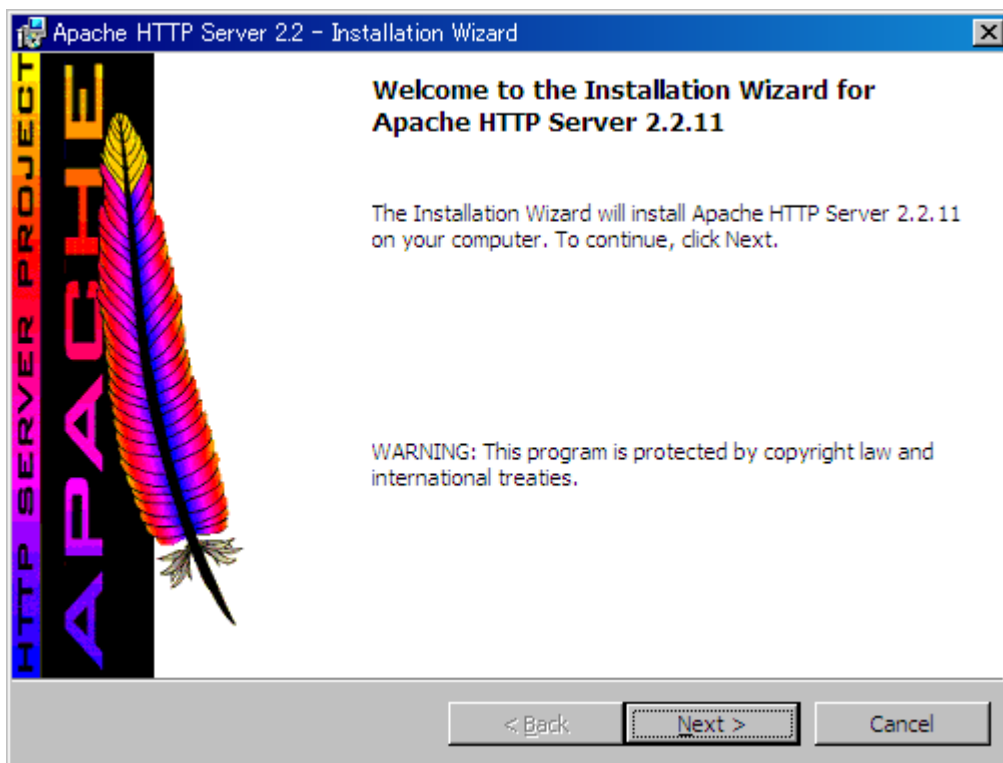
Index of /archives/net/apache/httpd/binaries/win32

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 HEADER.html	22-Mar-2007 12:03	631	
 LEGACY.html	22-Mar-2007 10:37	5.3K	
 README.html	15-Dec-2008 00:34	11K	
 TROUBLESHOOTING.html	22-Mar-2007 10:37	2.7K	
 apache_2.0.63-win32-x86-no_ssl.msi	19-Jan-2008 10:13	4.2M	
 apache_2.0.63-win32-x86-no_ssl.msi.asc	19-Jan-2008 10:13	481	
 apache_2.0.63-win32-x86-openssl-0.9.7m.msi	19-Jan-2008 10:17	4.7M	
 apache_2.0.63-win32-x86-openssl-0.9.7m.msi.asc	19-Jan-2008 10:17	481	
 apache_2.2.11-win32-x86-no_ssl.msi	13-Dec-2008 23:37	4.6M	
 apache_2.2.11-win32-x86-no_ssl.msi.asc	13-Dec-2008 23:37	827	
 apache_2.2.11-win32-x86-openssl-0.9.8i.msi	13-Dec-2008 23:37	5.2M	
 apache_2.2.11-win32-x86-openssl-0.9.8i.msi.asc	13-Dec-2008 23:37	827	
 patches applied/	16-Oct-2008 01:46	-	
 symbols/	15-Dec-2008 01:07	-	

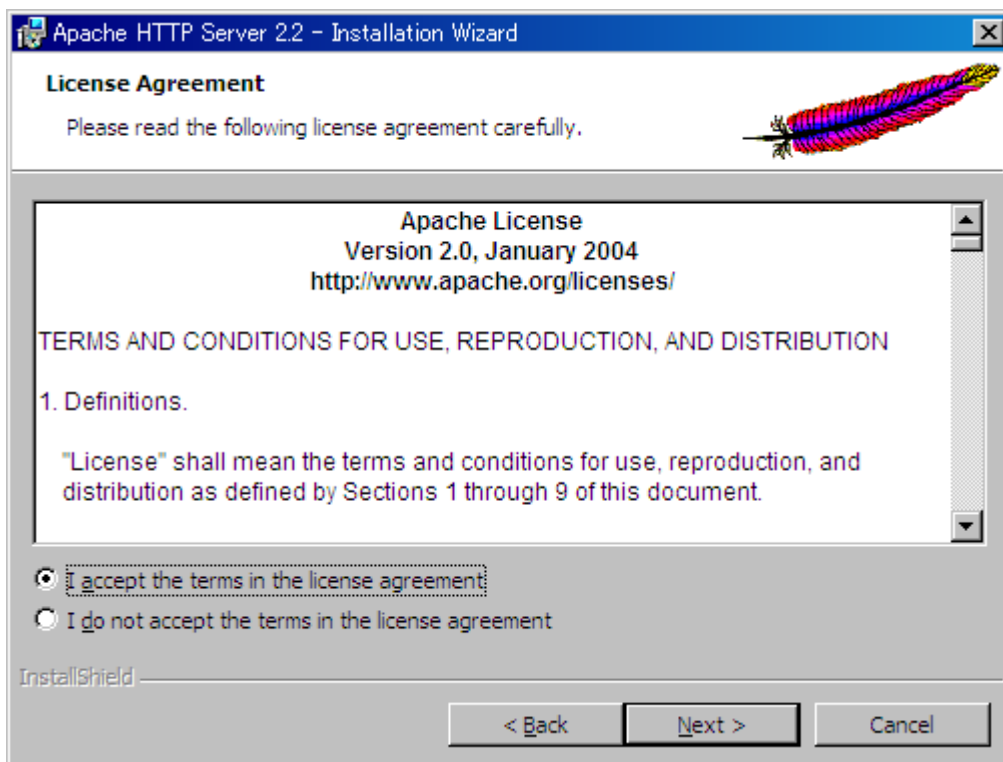
Download from your [nearest mirror site!](#)

Please do not download from www.apache.org. Use a mirror site to help us save apache.org bandwidth and to speed up your download. [Click here](#) to find your

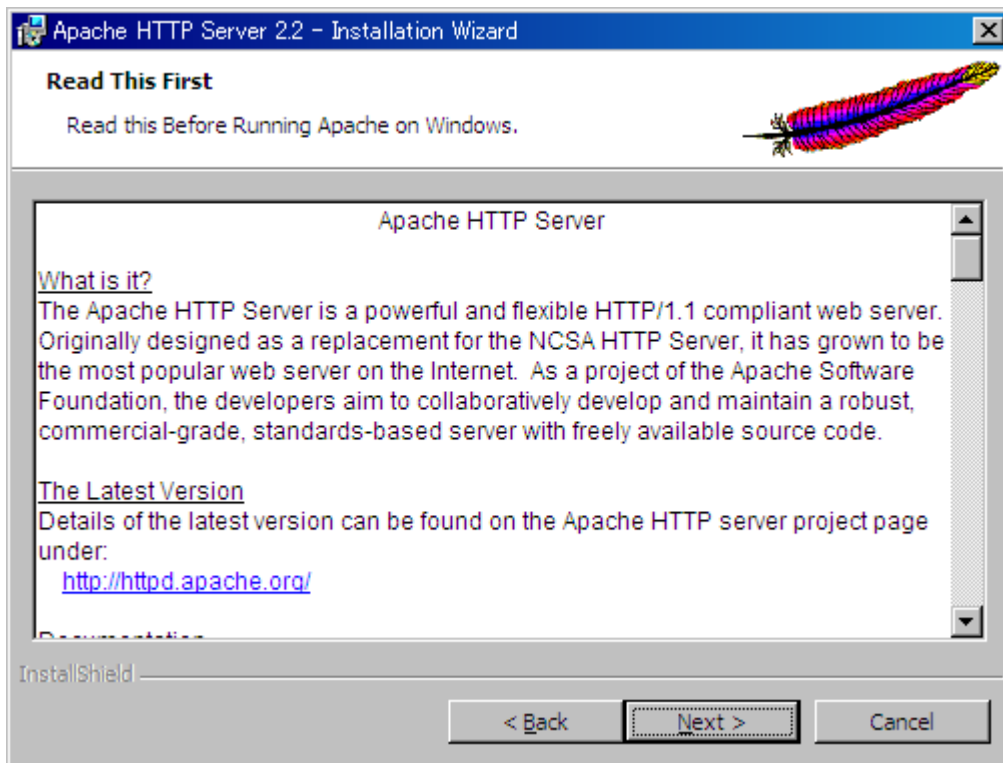
apache_2.2.11-win32-x86-openssl-0.9.8i.Msi
を選択しダウンロードしインストールします。



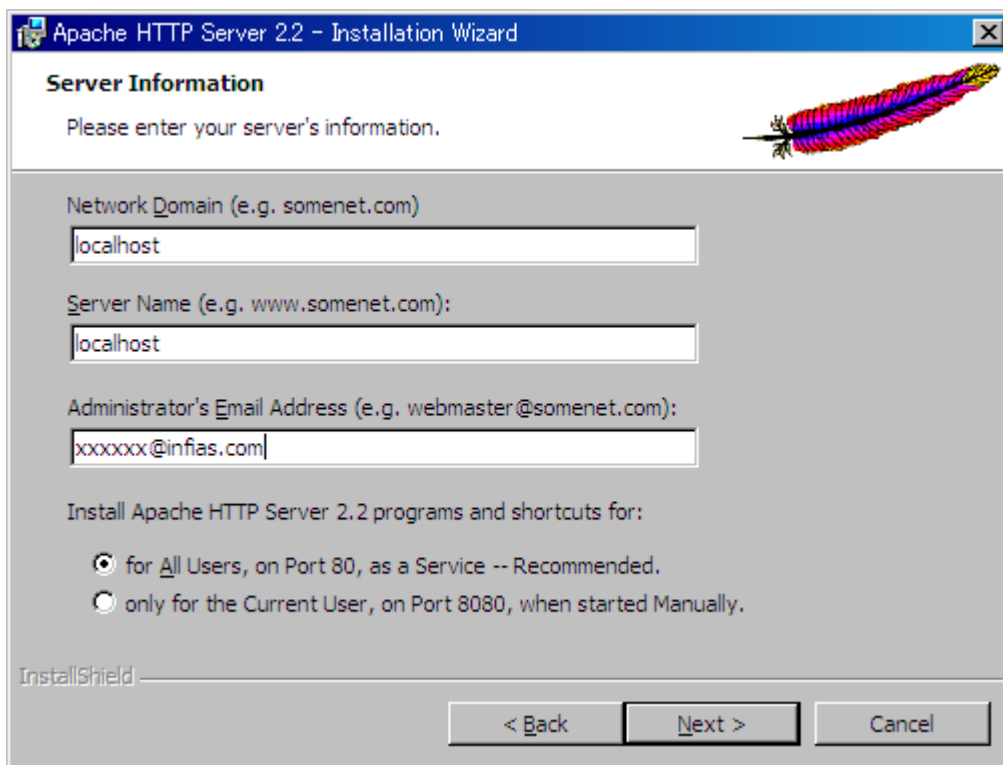
Next をクリック



I accept the terms in the license agreement をチェックし Next

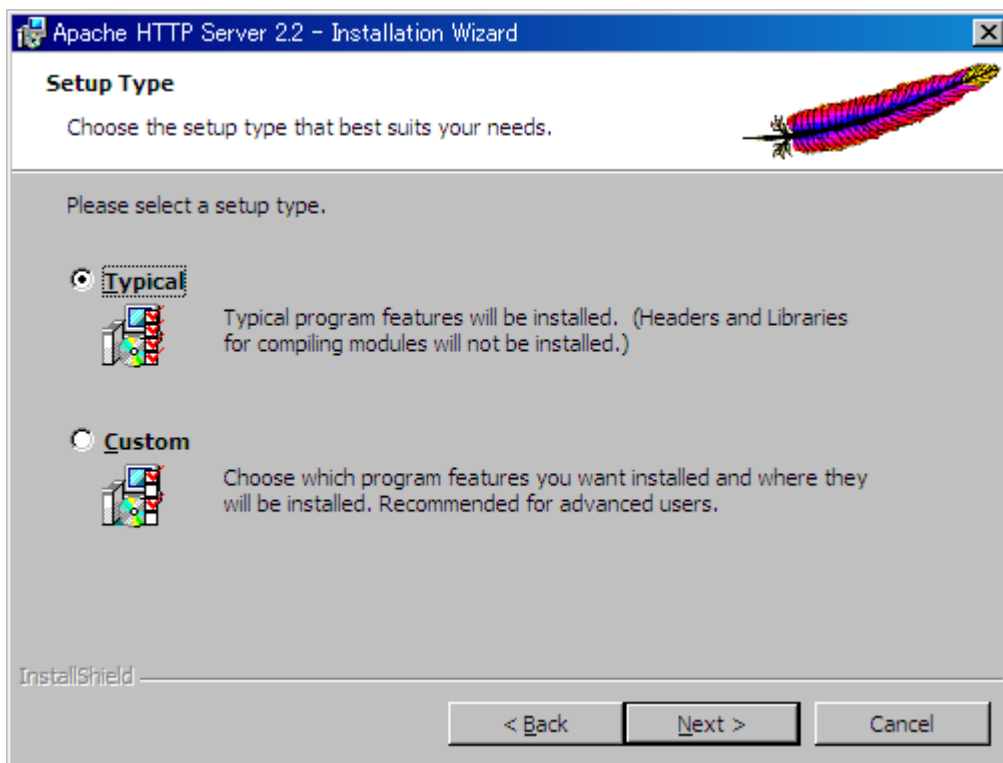


Next

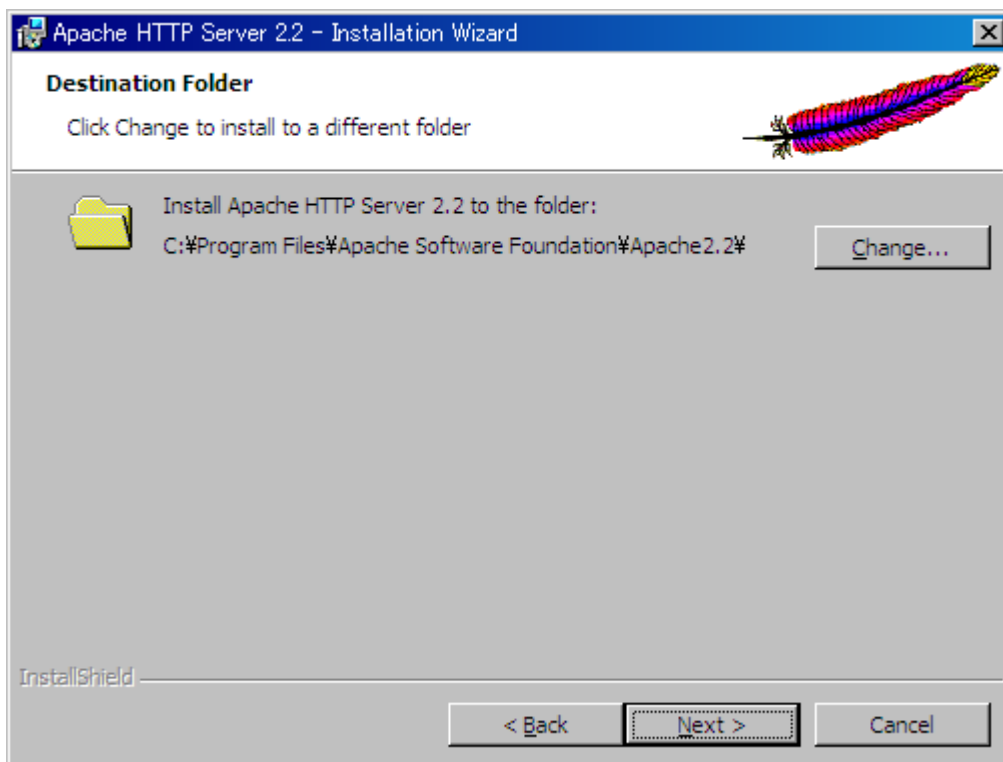


サーバーの情報を入力します。

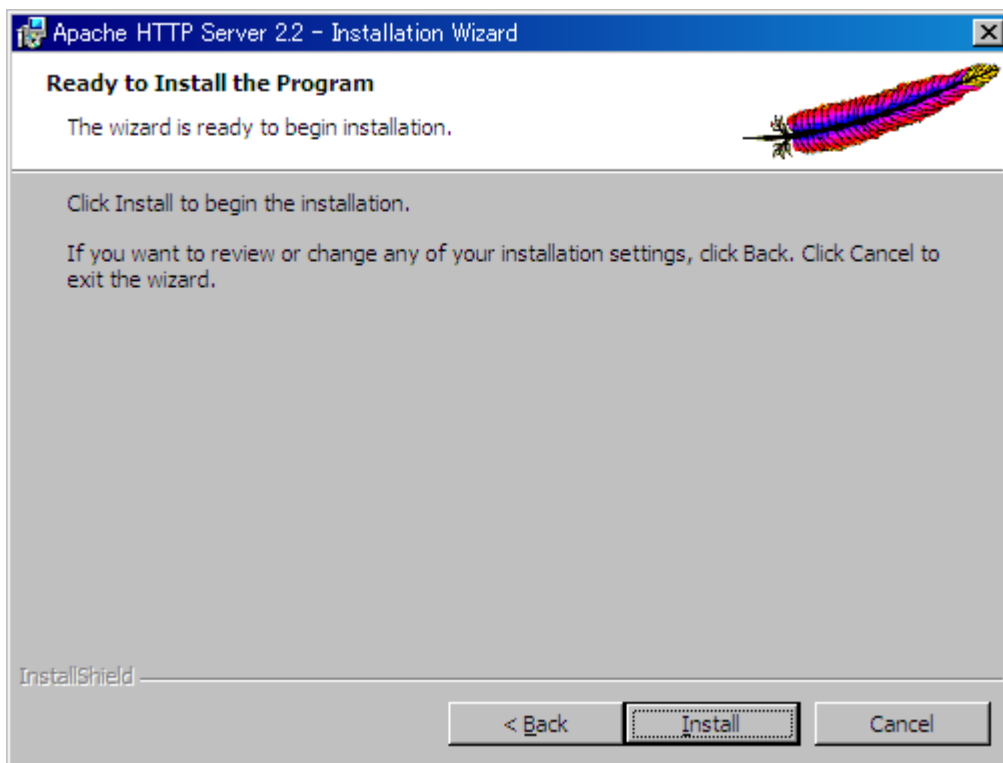
- Network Domain
サーバーのドメインを指定します。
ここでは「localhost」とします。
- Server Name
サーバー名を指定します。
ここでは「localhost」とします。
- Email Address
自分のメールアドレスを入力します。



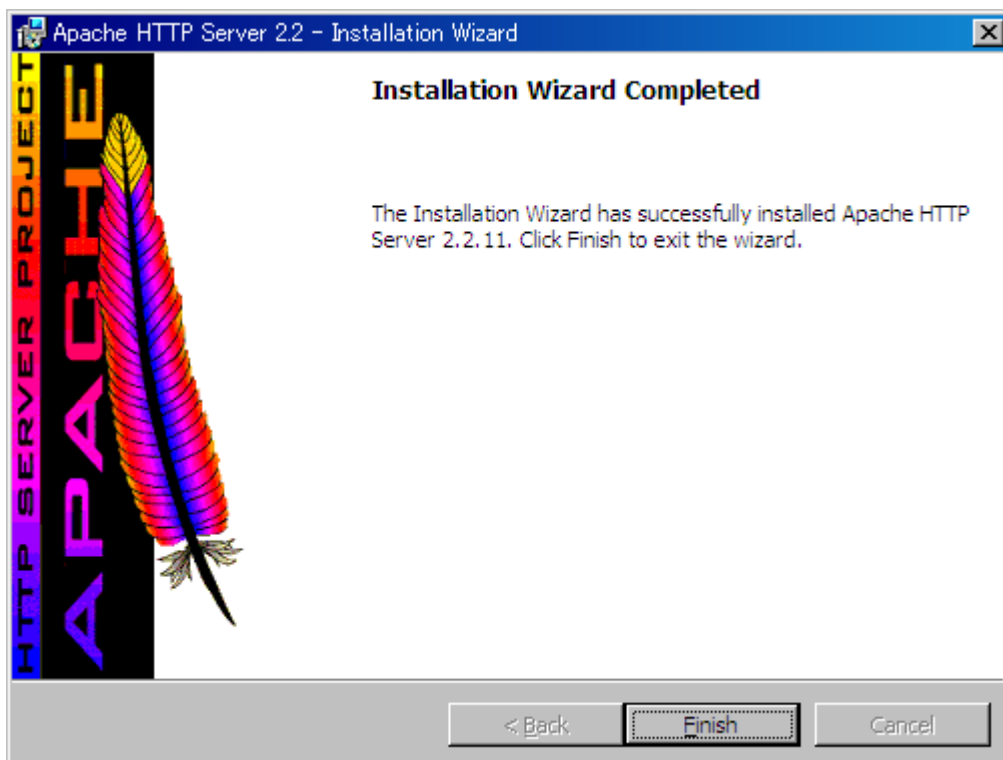
Typical を選択



Next



Install

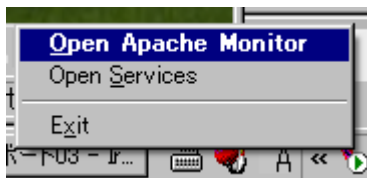


無事インストールが完了しました。
この状態ですすでにアパッチは起動しています。

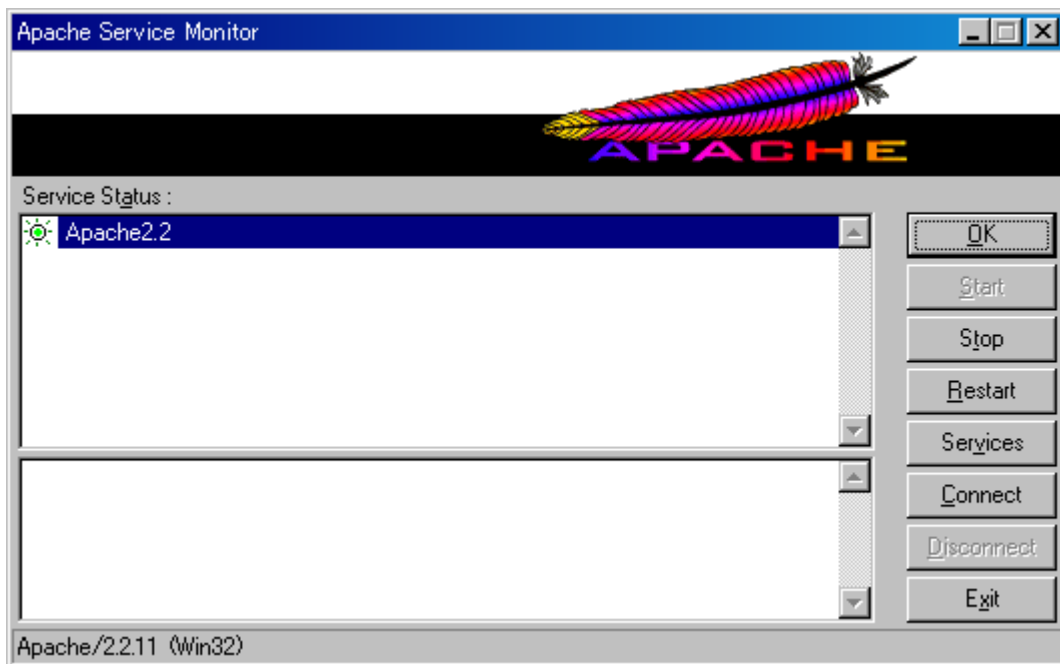


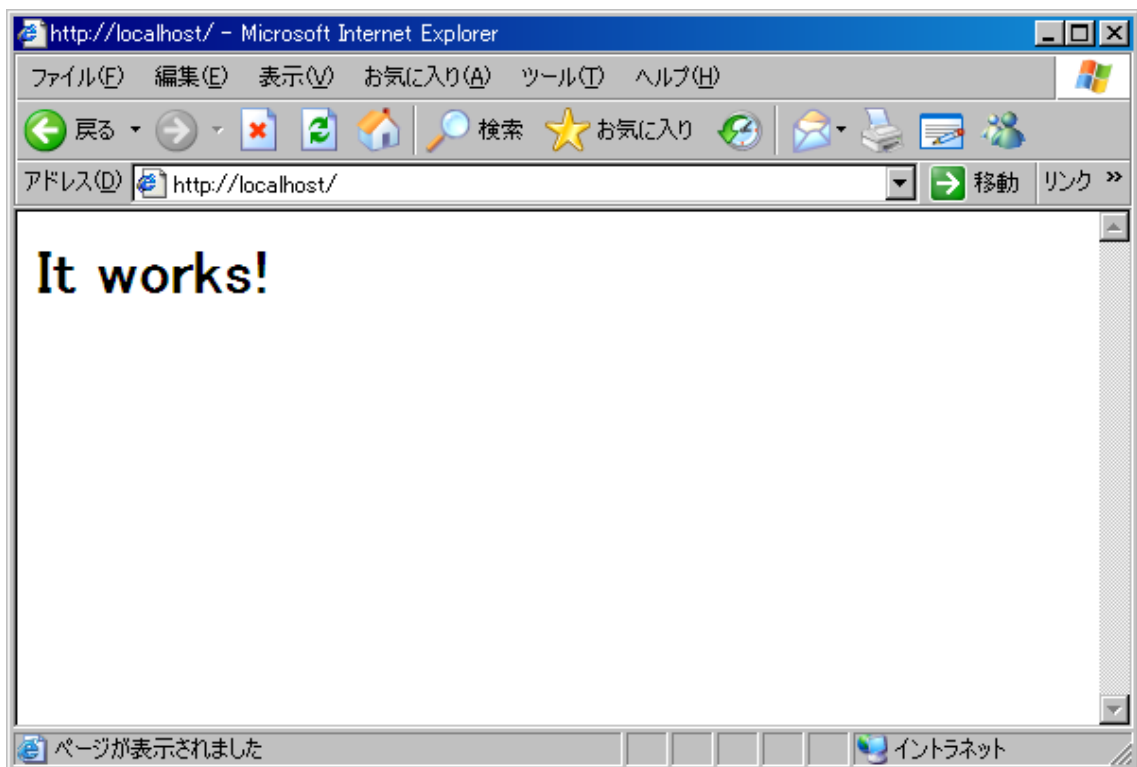
ちなみにアパッチのストップ、リセット、スタートはメニューから選択できます。

また、アパッチでは設定ファイルを書き換えたそのつどにリスタートしないと設定が反映されませんのでこの点は必ず覚えておきましょう。設定ファイルの変更などは後ほど。



起動するとタスクバーに上記のようなアイコンが表示されます。ここからも再起動などの設定が可能です。再起動は Restart もしくは、Stop をおしてあらためて Start を押します。





Apache が起動しているのか確かめてみましょう。

ローカルホストに設定した、<http://localhost/> または、<http://127.0.0.1/> と打ってみて上記のような画面が表示されれば Apache のインストールは完了です。

4.PHP のインストール

PHP は Apache 同様オープンソースであり誰でも自由に入手することができます。
まずは、<http://www.php.net/downloads.php> にアクセスします。



The screenshot shows the PHP website's download page for version 5.2.9. The page has a purple header with the PHP logo and navigation links: downloads, documentation, faq, getting help, mailing lists, licenses, wiki, reporting bugs, and a search bar. The main content is divided into two columns. The left column is titled 'Binaries for other systems' and contains text explaining that binaries are not distributed for UNIX/Linux, along with a list of supported operating systems: AS/400, Mac OS X, Novell NetWare, OS/2, RISC OS, SGI IRIX 6.5.x, Solaris (SPARC, INTEL), and Solaris OpenCSW packages. Below this is a section for 'Development and archive versions'. The right column is titled 'PHP 5.2.9 Complete Source Code' and lists two download options: 'PHP 5.2.9 (tar.bz2)' [9,964Kb] - 26 February 2009 and 'PHP 5.2.9 (tar.gz)' [12,566Kb] - 26 February 2009. Below this is a section for 'Windows Binaries' which lists three download options: 'PHP 5.2.9-1 zip package' [10,272Kb] - 10 March 2009, 'PHP 5.2.9-1 installer' [17,304Kb] - 10 March 2009, and 'PHP 5.2.9 Win32 Debug Pack' [5,172Kb] - 10 March 2009. A note states that the PECL package will not be released for this version. The third download option is 'PHP 5.2.9-1 Non-thread-safe zip package' [10,178Kb] - 10 March 2009.

Windows Binaries の PHP 5.2.9-1 zip package をダウンロードします。
ダウンロードサーバを聞かれるので日本を選択してファイルを落とします。
解凍したらフォルダごと C ドライブの直下などわかりやすいところに配置します。(以降 C ドライブ直下が前提)

そうしたら、解凍されたファイルの中から「php5ts.dll」「libmysql.dll」を Windows のシステムフォルダである「C:/Windows/system32」(Windows NT の場合は「C:/Winnt/system32」) にコピーします。また、同じく「php.ini-dist」を、「C:/Windows」(Windows NT の場合は「C:/Winnt/system32」) にコピーして、「php.ini」に名前を変更します。

以降、PHP の設定は、「C:/Windows/system32」配下の「php.ini」をテキストエディタで開いて設定を行います。また、設定後はアパッチの再起動をしてあらためて設定変更が反映されるので、PHP の設定を行った際はアパッチの再起動をしてください。

5.PHP の環境設定 (PHP.ini の設定)

「php.ini」を開いて、以下のとおりに設定を変更します。
インクルードパスのコメント記号(この場合はセミコロン)をはずし、パスを指定します。require(),include()などを使用する場合に必要となります。今回は Windows が対象なので Windows 下の先頭にあるセミコロンをはずします。
UNIX 系対象 OS では記述にあるとおり、対象のものを変更しましょう。設定に関しては Windows、UNIX ともにファイルのあるパス以外は共通なので覚えておきましょう。

```
469 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; ←
470 ; Paths and Directories ; ←
471 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; ←
472 ←
473 ; UNIX: "/path1:/path2" ←
474 ;include_path = "./:/php/includes" ←
475 ; ←
476 ; Windows: "%path1;%path2" ←
477 ;include_path = ".:c:%php%includes" ←
```

ドキュメントルートへのパスを設定します。

ドキュメントルートとは実際に公開され、外部のネットワークからみられる Web ページのあるところです。通常はアップパッチのインストールされたフォルダ配下の htdocs フォルダをしています。それと、設定ファイルで使うスペース記号は半角のものを 사용합니다。

```
479 ; The root of the PHP pages, used only if nonempty. ←
480 ; if PHP was not compiled with FORCE_REDIRECT, you SHOULD set doc_root ←
481 ; if you are running php as a CGI under any web server (other than IIS) ←
482 ; see documentation for security issues. The alternate is to use the ←
483 ; cgi.force_redirect configuration below ←
484 doc_root = "C:\Program Files\Apache Software Foundation\Apache2.2\htdocs" ←
```

拡張モジュールへのパスを変更します。

```
490 ; Directory in which the loadable extensions (modules) reside. ←
491 extension_dir = "C:\php\ext" ←
```

php_mbstring.dll のコメント記号を外します。

```
622 ;extension=php_ldap.dll ←
623 extension=php_mbstring.dll ←
624 ;extension=php_mcrypt.dll ←
```

次に、「extension=」が並んでいる場所に、以下のように追加します。MySQL のバージョンが 4.1.2 未満である場合は、改良版の MySQL 拡張サポートは動作しませんので、追加を行わずに、「extension=php_mysql.dll」のコメント記号を外してください。

```
629 ;extension=php_mssql.dll ←
630 extension=php_mysql.dll ←
631 ;extension=php_mysqli.dll ←
```

SJIS ユーザ用の設定を行います。PHP スクリプトの内部エンコードが SJIS、Web ブラウザ出力も SJIS である場合の設定方法です。

```
100 ; a value for this directive (e.g., output_buffering=4096). ←
101 output_buffering = 0 ←
```

出力バッファリングを有効にします。

```
114 ; Instead you must use zlib.output_handler. ←
115 output_handler = mb_output_handler ←
```

指定した関数を使用して、スクリプトの出力を行います。指定した場合、出力バッファリングは自動的に有効になります。

```
462 default_mimetype = "text/html"↵
```

```
463 default_charset = "Shift_JIS"↵
```

デフォルトの文字コードを Shift_JIS にします。

```
1188 [mbstring]↵
```

```
1189 ; language for internal character representation.↵
```

```
1190 mbstring.language = Japanese↵
```

デフォルトの言語を日本語に設定します。

```
1192 ; internal/script encoding.↵
```

```
1193 ; Some encoding cannot work as internal encoding.↵
```

```
1194 ; (e.g. SJIS, BIG5, ISO-2022-*)↵
```

```
1195 mbstring.internal_encoding = SJIS↵
```

内部エンコーディングを設定します。この設定にあわせた文字コードで PHP スクリプトを保存しなければなりません。

```
1197 ; http input encoding.↵
```

```
1198 mbstring.http_input = auto↵
```

HTTP 入力文字コードを設定します。auto を指定すると自動変換されます。

```
1200 ; http output encoding. mb_output_handler must be↵
```

```
1201 ; registered as output buffer to function↵
```

```
1202 mbstring.http_output = SJIS↵
```

Web ブラウザに出力する文字コードを設定します。

```
1204 ; enable automatic encoding translation according to↵
```

```
1205 ; mbstring.internal_encoding setting. Input chars are↵
```

```
1206 ; converted to internal encoding by setting this to On.↵
```

```
1207 ; Note: Do _not_ use automatic encoding translation for↵
```

```
1208 ; portable libs/applications.↵
```

```
1209 mbstring.encoding_translation = On↵
```

入力文字のエンコーディング検出および内部文字エンコーディングへの変換を行います。

```
1211 ; automatic encoding detection order.↵
```

```
1212 ; auto means↵
```

```
1213 mbstring.detect_order = auto↵
```

文字コード変換の優先順位を指定します。auto を指定すると自動設定されます。

```
1215 ; substitute_character used when character cannot be converted↵
```

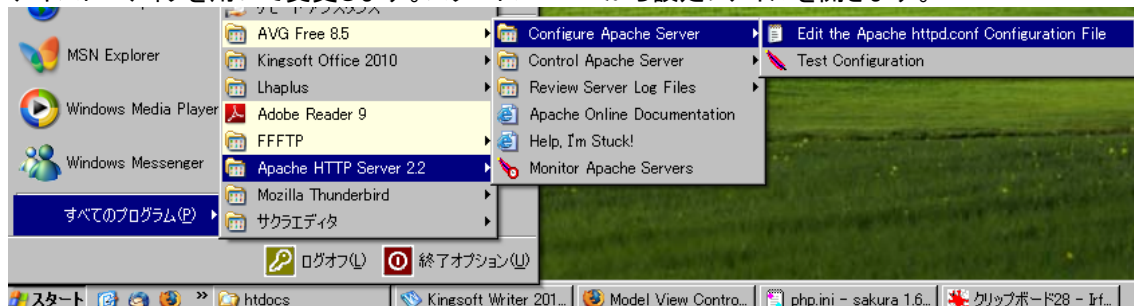
```
1216 ; one from another↵
```

```
1217 mbstring.substitute_character = none;↵
```

代替文字のデフォルト値を指定します。

6.Apache の設定 (httpd.conf の設定)

アパッチの設定を行います。アパッチも PHP と同様、設定ファイルはテキスト形式により記述されていますので、テキストエディタを用いて変更します。スタートメニューから設定ファイルを開きます。



```
57 # Example: ←  
58 # LoadModule foo_module modules/mod_foo.so ←  
59 # ←  
60 ←  
61 LoadModule php5_module "C:%php%php5apache2_2.dll" ←  
62 LoadModule actions_module modules/mod_actions.so ←
```

ファイルを開いたら、「LoadModule」が連続するあたりに、マーカーのついている行を挿入してください。なお、PHP の設定ファイルではセミコロンがコメント扱いでしたがアパッチの設定ファイルではシャープがコメント扱いとなっております。

```
131 #insert 09/04/07 ←  
132 AddDefaultCharset Off ←
```

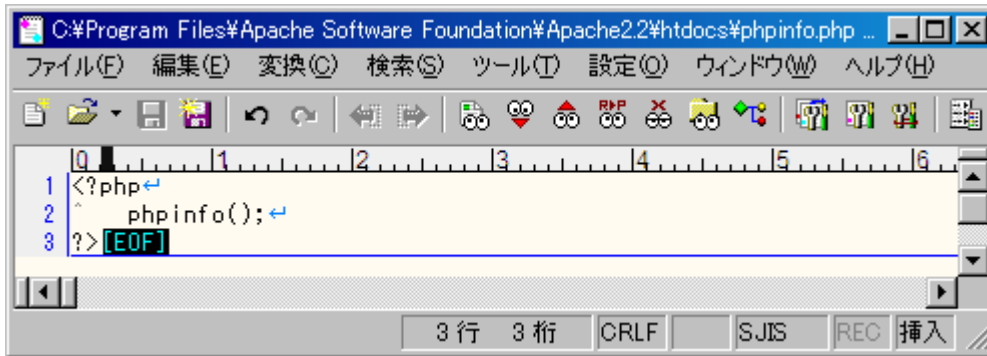
AddDefaultCharset Off 行を追記します。

```
386 AddType application/x-compress .Z ←  
387 AddType application/x-gzip .gz .tgz ←  
388 AddType application/x-httpd-php .php ←
```

AddType が連続する箇所の所に行を追加してください。設定の変更を反映するために、アパッチを再起動してください。以上でアパッチの基本的な設定は終わりです。おつかれさまでした。

7.PHP 動作確認

phpinfo.php というファイルをつくってみましょう。それを、アパッチの公開フォルダに入れて開いてみます。このときファイルの拡張子は「.php」という名前です。



```
1 <?php
2     phpinfo();
3 ?>
```

ブラウザで開いてみます。アドレスは、<http://localhost/phpinfo.php> です。



System	Windows NT INFIAS-TEMP0001 5.1 build 2600
Build Date	Mar 5 2009 20:01:54
Configure Command	cscrip /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-snapshot-template=d:\php-sdk\snap_5_2\vc6\template" "--with-php-build=d:\php-sdk\snap_5_2\vc6\php_build" "--with-pdo-oci=D:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8=D:\php-sdk\oracle\instantclient10\sdk,shared" "--enable-htscanner=shared" "--without-pi3web"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded	C:\WINDOWS\php.ini

このような画面が表示されればインストール成功です。見れない場合は設定を見直したりアパッチの再起動を確認しましょう。アパッチと PHP のインストールと環境設定はここまでです。おつかれさまでした。

8.Linux 環境下での LAMP 構築

サーバー構築にあたっては、Windows 系列の OS に比べ実は UNIX 系列の OS のシェアの方が多分を占めているのが現状です。実際にサーバー構築する際は UNIX 系列での構築を前提とすることがあるのでその方法を明記しておきます。UNIX とはもともとベル研究所というアメリカの NTT にあたるところが開発した OS でサーバーに特化した OS として知られています。現在では、同じような機能を持つ UNIX 互換 OS と呼ばれるものが普及しています。Linux もそのひとつで、他には、Solaris、BSD、Mac OS X などが UNIX 互換 OS として知られています。今回はそのなかの Linux を題材に進めていきます。基本的に UNIX 互換 OS での操作はターミナル、または端末と呼ばれる Windows でいうコマンドプロンプトのような CUI を使ったもので操作することになります。

LAMP(Linux,Apache、MySQL、PHP)環境の構築にあたってはソースからコンパイルしてインストールする方法と、パッケージと呼ばれるあらかじめバイナリ化したものを用いてインストールする方法の2パターンあります。また、パッケージにも大きく分けて Red Hat 系のものと、Debian 系のものにわかれます。以降からは Red Hat 系を前提としたソースからのインストールを紹介します。なを、**LAMP 構築にあたっては、必ず、MySQL から先にインストールしてください。**

➤ Apache のインストールと起動

あらかじめパッケージとしてインストールされているか確認します。パッケージ管理はルートにならないと確認できないのでルート権限になって打ちます。

✓ Apache のインストール状況の確認

```
# yum list installed | grep httpd  
あった場合、削除  
# yum remove httpd
```

<http://httpd.apache.org/download.cgi> から最新版のソースを落とします。

適当な場所にダウンロードして解凍

```
# tar xvzf httpd-2.2.11.tar.gz
```

作成されたディレクトリに移動

```
# cd httpd-2.2.11
```

次にコンパイルのための設定を行います。--enable-so(DSO オプション)をつけてインストールすると、Apache をリコンパイルすることなく設定ファイルを書き換えるだけで動的にモジュールの追加や削除ができます。

```
# ./configure --enable-so
```

終わったら、コンパイルしてインストールします。

```
# make
```

```
# make install
```

これでインストールは終了です。エラーが出る場合はコンパイルの際に必要なファイルが存在していない場合があるのでそこはエラーコードを参照にパッケージでインストールします。

yum install エラーコードを参照に必要なパッケージの名前

Apache の起動

```
# /usr/local/apache2/bin/apachectl start
```

Apache の再起動

```
# /usr/local/apache2/bin/apachectl restart
```

Apache の停止

```
# /usr/local/apache2/bin/apachectl stop
```

Web ブラウザを起動させ、アドレス欄に <http://localhost/> もしくは <http://127.0.0.1/> を入力して Apache の初期画面が表示されれば起動確認は終了です。

➤ PHP のインストールと設定

✓ PHP のインストール状況の確認

```
# yum list installed | grep php
あった場合、削除
# yum remove php
```

✓ PHP のインストール

<http://www.php.net/downloads.php> から最新版のソースを落とします。
Apache と同じように解凍、ディレクトリへ移動

```
# tar xvfz php-5.2.9.tar.gz
# cd php-5.0.3
```

コンパイルのための設定を行います。インストールした MySQL のバージョンが 4.1.2 未満の場合は、改良版の MySQL 拡張サポートが動作しないので、`--with-mysqli` オプションを外します。

```
# ./configure ¥
--with-apxs2=/usr/local/apache2/bin/apxs ¥
--with-mysqli=/usr/local/mysql/bin/mysql_config ¥
--with-mysql=/usr/local/mysql ¥
--enable-mbstring ¥
--enable-mbregex
```

```
# ./configure 改行
PHP を Apache の DSO 拡張モジュールとして動作させます 改行
PHP の mysqli 拡張サポートを有効にします 改行
PHP の mysql 関数を有効にします 改行
mbstring(マルチバイト拡張)関数を有効にします 改行
マルチバイトに対応した正規表現関数を有効にします
```

設定と設定の説明は以上です。
次にコンパイルしてインストールします。

```
# make
# make install
```

php.ini-dist を php.Ini の名前で /usr/local/lib にコピーします。
cp php.ini-dist /usr/local/lib/php.Ini

✓ php.ini の設定

/usr/local/lib/php.ini を開いて、以下のとおりに設定を変更します。

インクルードパスを変更する。

```
include_path = ".:usr/local/include/php:usr/local/lib/php"
```

ドキュメントルートへのパスを設定します。

```
doc_root = "/usr/local/apache2/htdocs"
```

拡張モジュールへのパスを変更します。

```
extension_dir = "/usr/local/include/php/ext"
```

EUC-JP ユーザ用の設定を行います。次の設定は、PHP スクリプトの内部エンコードが EUC-JP、Web ブラウザ出力も EUC-JP である場合の設定方法です。以下を参考に、php.ini 内の設定項目を探して変更してください。コメントアウトされているものについては、コメント記号を外してください。

出力バッファリングを無効にします。

```
output_buffering = off
```

デフォルトの文字コードを EUC-JP にします。

```
default_charset = "EUC-JP"
```

デフォルトの言語を日本語に設定します。

```
mbstring.language = Japanese
```

内部エンコーディングを設定します。

(この設定にあわせた文字コードで PHP スクリプトを保存しなければなりません)

```
mbstring.internal_encoding = EUC-JP
```

HTTP 入力文字コードを設定します。auto で自動変換を行います。

```
mbstring.http_input = auto
```

Web ブラウザに出力する文字コードを設定します。

```
mbstring.http_output = EUC-JP
```

入力文字のエンコーディング検出および内部文字エンコーディングへの変換を設定します。

```
mbstring.encoding_translation = On
```

文字コード変換の優先順序を指定します。auto を指定すると自動設定されます。

```
mbstring.detect_order = auto
```

代替文字のデフォルト値を指定します。

```
mbstring.substitute_character = none;
```

➤ httpd.conf の設定

Apache の設定ファイルの編集を行います。

/usr/local/apache2/conf/httpd.conf を開いて、AddType の連続する箇所の下に以下を追加します。

```
AddType application/x-httpd-php .php
```

また、以下の行の ISO-8859-1 を Off に変更します。ファイルに存在しない場合には、この行を追加してください。

```
AddDefaultCharset ISO-8859-1
```

↓

```
AddDefaultCharset Off
```

以上を変更したら、変更内容を反映させるために、Apache を再起動します。

```
# /usr/local/apache2/bin/apachectl restart
```

➤ 動作確認

PHP の動作確認をするために Windows 編のときに作成した phpinfo.php と同じコードを作成して、/usr/local/apache2/htdocs に保存します。

Web ブラウザより以下のアドレスを入力して、PHP の情報を表示できれば成功です。

<http://localhost/phpinfo.php>

以上でインストールと環境設定は終わりですお疲れ様でした。