

## 環境の構築方法

### 1 Rubyのインストール

「One-Click Ruby Installer 186-xx」のファイナルリリースを入手します。

入手元

<http://rubyforge.org/projects/rubyinstaller/>

テストバージョン

ruby186-26.exe

インストーラを実行し、任意のフォルダにRubyをインストールします。  
このとき、コマンド実行が行い易い様に、ドライブ直下のフォルダにすると便利です。

例)

C:\Ruby

### 2 Railsのインストール

「1 Rubyのインストール」でRubyをインストールしたフォルダ下で下記のコマンドを実行します。

バージョンを指定してのインストール

```
C:\Ruby> gem install rails --version 2.1.2 --include-dependencies
```

最新版のインストール

```
C:\Ruby> gem install rails --include-dependencies
```

### 3 プラグインなどのインストール

「1 Rubyのインストール」でRubyをインストールしたフォルダ下で下記のコマンドを実行します。

```
C:\Ruby> gem install rails -v=2.1.2
C:\Ruby> gem install uuid -v=1.0.4
C:\Ruby> gem install reliable-msg -v=1.1.0
C:\Ruby> gem install ap4r -v=0.3.7
C:\Ruby> gem install gettext -v=1.93.0
C:\Ruby> gem install vpim -v=0.695
C:\Ruby> gem install mocha -v=0.9.5
C:\Ruby> gem install json -v=1.1.4
C:\Ruby> gem install mongrel -v=1.1.5
C:\Ruby> gem install fastercsv -v=1.4.0
C:\Ruby> gem install fastthread -v=1.0.7
C:\Ruby> gem install httpclient -v=2.1.4
C:\Ruby> gem install nokogiri -v=1.2.3
C:\Ruby> gem install soap4r -v=1.5.8
C:\Ruby> gem install actionwebservice
```

上記、プラグインをインストール際に、下記のインストールするタイプを選択するメッセージが表示されることがあります。

```
C:\Ruby>gem install gettext
Select which gem to install for your platform (i386-mswin32)
1. gettext 1.91.0 (ruby)
2. gettext 1.90.0 (ruby)
3. gettext 1.10.0 (mswin32)
4. gettext 1.10.0 (ruby)
5. Skip this gem
6. Cancel installation
```

Windows用のタイプの中で、バージョンの新しいものを選択して、次に進んでください。

#### 4 アプリケーションのインストール

任意のフォルダに配布されるアプリケーションをコピーします。

#### 5 Apache2. x のインストール

Apache2. x の最新版を入手します。

入手元

<http://httpd.apache.org/download.cgi>

テストバージョン

apache\_2.2.8-win32-x86-openssl-0.9.8g.msi

インストーラーを実行して、任意の場所にインストールを行います。

#### 6 データベース（MySQL編）をインストール

MySQLの最新版を入手します。

入手元

<http://dev.mysql.com/downloads/>

テストバージョン

mysql-5.0.51a-win32.zip

インストーラーを実行して、任意の場所にインストールを行います。

「MySQL Command Line Client」を利用して、MySQLにログインしデータベースを作成します。

```
mysql> create database データベース名 default character set utf8;
```

## 7 アプリケーションのデータベース設定（MySQL編）を変更します。

「4 アプリケーションのインストール」をコピーしたフォルダ下、configフォルダの下に「database.yml」を編集します。

database.yml

```
production[development/test]:  
  adapter: mysql  
  database: データベース名  
  username: ユーザ名  
  password: パスワード  
  host: サーバ名  
  encoding: utf8
```

### 注意

MySQL以外のデータベースは、データベース接続アダプタをインストールする必要があります。データベース毎に設定する内容が違います、かならずリファレンス等をお読みください。Windows機でPostgreSQLを利用する場合は「PostgreSQL8.2.5」を推奨します。

## 8 アプリケーションに必要なスキーマをデータベースに作成する。

「4 アプリケーションのインストール」をコピーしたフォルダの下で、下記のコマンドを実行します。

```
C:\Pjc3> rake makemo  
C:\Pjc3> rake db:migrate VERSION=0  
C:\Pjc3> rake db:migrate  
C:\Pjc3> rake db:sic:load  
C:\Pjc3> rake gettext_from_db
```

## 9 ApacheとMongrelを接続する設定を行います。

「5 Apache2. x のインストール」でApacheをインストールしたフォルダ下、confフォルダの下に「httpd.conf」を編集します。

LoadModuleの変更

下記のモジュールのコメント化を無効にします。

```
LoadModule proxy_module modules/mod_proxy.so  
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so  
LoadModule proxy_http_module modules/mod_proxy_http.so  
LoadModule status_module modules/mod_status.so
```

ロードバランス管理画面の設定

下記の内容を追記します。

```
<Location /balancer-manager>  
  SetHandler ロードバランス管理画面のURL (例 : balancer-manager)  
  Order Deny,Allow  
  Deny from all  
  Allow from 管理画面を使用するクライアントやネットワークを指定  
  (例 : 192.168.0. )  
</Location>
```

リバースプロキシ、ロードバランスの設定

下記の内容を追記します。

```
#フォワードプロキシ機能を無効にします。  
ProxyRequests Off
```

```
<Proxy *>  
  Order deny,allow  
  Allow from all  
</Proxy>
```

リバースプロキシの設定  
下記の内容を追記します。

```
ProxyPass / balancer://mongrel/ lbmethod=bytraffic timeout=1  
ProxyPassReverse / balancer://mongrel/  
<Proxy balancer://mongrel>  
BalancerMember Mongrel/サーバURL (ポート付) loadfactor=10  
</Proxy>
```

#### 注意

lbmethod、timeout、loadfactorは今後も調整が必要  
BalancerMemberは、複数のMongrelを登録する事が可能。

### 10 Mongrelのサービス化する方法

「1 Rubyのインストール」でRubyをインストールしたフォルダ下で下記のコマンドを実行します。

```
C:¥Ruby> gem install mongrel_service
```

「4 アプリケーションのインストール」をコピーしたフォルダの下で、下記のコマンド(実際には1行)を実行します。

```
C:¥Ruby> mongrel_rails service::install -N "サービス名"  
          -c アプリケーションのインストールフォルダ  
          -p ポート -e 起動モード
```

「-e」の「起動モード」には、下記のモードが存在します。

development 開発モード  
production 運用モード  
test テストモード

マスタメンテナンスツールを利用する際は必ず開発モードを利用します。