ToDo

2006-11-3

thumbnail プラグイン

とりあえずひっそりと上げてみる。

■特徴

・リサイズ画像はキャッシュとしてユーザに見えない形で管理
・サイズに名前を付け管理画面で一括管理することにより、途中でのサイズ一括変更も簡単
・リサイズに Image-Thumbnail を使用することにより、GD/ImageMagick/Imager に対応

■インストール方法

このプラグインの使用するには GD/ImageMagick/Imager のいずれかをインストールして使用可能にしておかなければなりません。

1.zip ファイルを展開すると thumbnail ディレクトリができる
2.thumbnail ディレクトリの中にある lib/Image/Thumbnail.pm と tmpl/thumbnail.tmpl を、それぞれ fswiki の lib/Image と tmpl ディレクトリに移動する。
3. 残りの thumbnail ディレクトリをそのまま plugin ディレクトリに移す

2006-1-7

アップグレード補助スクリプト

・現状認識
  ・現状スクリプト位置検出
    ・親ディレクトリ以下自身を除くディレクトリから cgi 本体の存在場所を探す
    ・設定ファイルにて直接設定可
    ・設定ファイルにて最大検索階層を設定可
    ・設定ファイルにて cgi 本体のファイル名を設定可
  ・移行用ファイル位置検出
    ・自身のディレクトリ中に移行用のファイル一式を収めたディレクトリを探す
    ・設定ファイルにて直接設定可
    ・ディレクトリ名 wiki3_* を標準とする
・基本バージョン確認
  ・cgi 内部のバージョン記述から抽出
・健全性確認
  ・バージョンに対応したファイル名と md5 ハッシュ値のリストを取得
    ・wiki3_*.md5 をそのリストを保存したファイル名とする
  ・ファイル・ディレクトリ属性パターンの取得
  ・指定ディレクトリ以下を検索して問題点をチェックする
    ・ファイル過不足
    ・所有者
    ・パーミッション
    ・ファイル改変
    ・attach,pdf,docs は存在の報告のみ
    ・backup,log は存在する farm（サブディレクトリ）の報告のみ
    ・config,data は特定ファイルと存在する farm（サブディレクトリ）の報告のみ
  ・問題点の解析・評価・報告

2005-10-29

wiki ネタメモ：Graphviz

Graphviz による作図プラグイン。

・ブロックプラグインにより dot を書くと作図して表示する。
　・作った画像は添付ファイルへ
　　・ブラグインから添付ファイルを作る方法を調べる必要あり
　・プラグイン自体は添付ファイルを表示する ref_image プラグイン記述を返す
・添付した dot を作図して画像として表示する ref_graphviz プラグイン
　・on the fly だと重くなるかも知れないので画像はキャッシュする

2005-10-23

サブスタイル設定プラグイン

追加のスタイルシートと alternative stylesheet を追加するプラグインを作ってみた。とりあえず機能は低めだけど、叩き台的になればという事で公開してみる。

・http://aaa-www.net/~typer/substyle.tar.gz

インストール方法は、

・上記ファイルを展開して出来たファイルをアップロード
・プラグイン設定で substyle プラグインを有効にする
・スタイル設定でサイトテンプレートを substyle にする
・サブスタイル設定で追加のスタイルシートや alternative にするテーマを設定する

といった所。

サイトテンプレートの変更は、標準のスタイルシートの挿入場所と、title 属性がついていないために、このプラグインが期待通りの動作をしないために必要になった。変更点は少ないので、default サイトテンプレート以外を使用している場合も、同様の変更を加えれば使える。

追加のスタイルシートは、標準では theme/substyle ディレクトリに入れれば使えるようになる。このディレクトリも変更する事が出来る（サブスタイル設定の 3 つ目の設定項目）。

2005-10-16

code プラグインに行番号を引き継げる様にするパッチを作った

これ作っている時に思いついたネタとして、こういう diff/patch を色分けするようなのもいいかなと思った。

```
--- Code.pm.orig  Sun Sep  4 03:14:16 2005
+++ Code.pm  Sun Oct 16 13:27:02 2005
@@ -22,6 +22,15 @@
    return bless $self,$class;
 }

+use vars qw($maxline $place);
+#==========================================================
+# フックメソッド（initialize用）
```

```
+#=======================================================
+sub hook {
+  $maxline = 1;
+  $place = 1;
+}
+
 #=======================================================
 # ブロックプラグイン
 #=======================================================
@@ -62,8 +71,16 @@
     # 行番号表示
     if    ($line eq 'false') {$line='0';}
     elsif ($line eq 'true' || $line =~ /^ *$/ ) {$line='1';}
-    if ($line !~ /[^ 0-9]/ && $line !~ /^ *0$/){
-      my $maxline = split(/\n/,$source) + $line-1;
+    if ($line !~ /[^-+ 0-9]/ && $line !~ /^ *0$/){
+      if ($line =~ /[+-]/){
+        if ($line < 0 or $line =~ /\+/ and $line == 0){
+          $line++;
+        }
+        $line = $maxline+$line;
+        $line = " "x($place - length $line) . $line;
+      }
+      $place = length $line;
+      $maxline = split(/\n/,$source) + $line-1;
       $html .= "<td class=\"line\"><pre>";
       for(my $i=$line;$i<=$maxline;$i++){
         $html .= "$i<br>";
--- Install.pm.orig  Sun Oct 16 01:07:16 2005
+++ Install.pm  Sun Oct 16 13:26:21 2005
@@ -10,6 +10,7 @@
   my $wiki = shift;
   $wiki->add_block_plugin("code","plugin::code::Code","HTML");
   $wiki->add_paragraph_plugin("ref_code","plugin::code::RefCode","HTML");
+  $wiki->add_hook("initialize","plugin::code::Code");
 }

 1;
```

2005-10-13

block プラグイン差し替え

block プラグインのためのパッチが、プラグイン版になったためここに導入してみた。

兄弟 wikifarm、code:FrontPage で code プラグインを使用しているため、その wikifarm のみで有効にし、これまでの block プラグインのためのパッチがあたっていた部分を、あたっていない物に置き換えた。通常ならばパッチがあたっていても害はないが、ここではテストもかねているため、念のためにあたっていない物に差し替えている。ただし、元に戻したファイルの内、「lib/Wiki.pm」はオリジナルの物になったが、それ以外の「lib/Wiki/HTMLParser.pm」と「lib/Wiki/Parser.pm」は、先の Hatenize プラグイン関連でパッチがあたっていたため、オリジナルにパッチをあて直した物で差し替えている。

さて、ここで一つ気が付いた事がある。今回の場合は、Hatenize 用のスタイル関連のパッチと、_ex_block プラグインは干渉し合う事なく共存できているが、そうでない場合で farm を運営する場合は、パッチを本体とプラグインの双方にあてなければならないかもしれない。なぜならば、プラグインを有効にしている farm と無効にしている farm では走るソースの場所が違うからである。ただ、たとえばプラグインを有効にする事でパッチが無効となる事を承知の上ならば問題ない。

このことは、逆に、パッチでは、これまで farm 間で有効無効が変えられなかったが、そのパッチをプラグイン方式に変えれば farm 間でも有効無効を変えられるようになる、つまり自由度があがるという事でもある。

2005-9-28

Hatenize プラグイン更新

幾つかバグフィクスした。今の所、バージョン番号はつけない。

2005-9-26

ToDo が全然変わってないんですけど

理由は思いついた事を次々やってたから。今思うに、それって夏休みモードなのかも。子どもの頃、夏休みって明日とか来週とか来月とか考えずに好きな事やってた気がする。それと CardForm から逃げてたのもある？おぼろげな構想はあってもまとまりきらず、置きっぱなしにしてる。そして？界隈でごそごそやってたから、ここの日記にも書かなかったと。

で、ようやく夏休みが終ったのか、ちょっと計画を建てて見ようと思った。そんなわけで、やる／やりたい事。ただし、順位が付いてないので ToDo には上げない。

- Hatenize プラグインをプラグイン投稿に投げる←ここ関係
  - ヘルプを整理する
  - ソースを code:FrontPage で解説＆整理する
- Regexp::Trie による正規表現最適化をキーワードリンクにも応用する←あるいみ本題
  - キーワード数や対象文字数と最適化効果の関係を調べる
- FreeBSD 6.0 リリースに向け、UPDATING を整理する← TranslateFreeBSDlogs 関係
- はてなアイデアの株売買動向調査← ?G 関係

2005-9-19

はてなキーワードへのリンクを張る「Hatenize」プラグイン完成

とりあえず完成して、ここへ仕込んでいる。リンク張りまくりだ（笑）

■入手

まず、http://aaa-www.net/~typer/hatenize.tar.gz と共に、http://www.dan.co.jp/~dankogai/cgi/hatenize/ から、hatena2list.pl と mk_trie_regexp.pl を入手します。hatenize.tar.gz を伸長すると、2 つのパッチと 3 つのプラグインが入ったディレクトリがあります。

■プラグイン設置

3 つのプラグインの内、anotherlink はキーワードパーサ乗っ取りと拡張を行ない、hatenize がその拡張機能によりはてなキーワードへのリンクを張るプラグインとなっています。まずはこの 2 つを設置して有効にしてください。この時点では今までと変わらないはずです。エラーがなくなにも変わらなければ成功です。testlink はテスト用のプラグインで、これを有効にすると、「testlink」に http://example.com/ へのリンクが張られるはずです。確認用にどうぞ。

■キーワード正規表現ファイルの作成

次にキーワード正規表現ファイルの作成を行ないます。この方法には 2 つあり、一つは cgi に行なわせる方法、もう一つは手元で作成してサーバに送る方法です。

cgi に作成させる場合は、fswiki のログファイルがあるディレクトリ（標準では「./log」）に「hatena.rx」というファイルサイズが 1 以上で 3 日以上前の更新日のファイルを作り、cgi にアクセ

スすれば作り始めます。ただし、この処理は非常に重く、私の環境（pen4 1.4GHz）で 1 分近くかかり、メモリもかなり消費しますので、環境によっては途中で強制終了をくらい、作る事が出来ないでしょう。強制終了をくらった場合、サーバにゴミファイルが出来ている可能性があります。先ほどの hatena.rx と同じ場所に数字のみのファイルが出来ていたら消しておいてください。また、hatena.rx の更新日時が更新されていると思いますが、中身は変わっていないと思います。再度挑戦する場合は、更新日時を 3 日以上前にすれば良いです。ただし、やりすぎるとサーバ管理者に文句をいわれますからほどほどに。

さて、手元で作る場合、さっき取得した hatena2list.pl に hatena2list.pl.patch というパッチをあてます。もともとの hatena2list.pl は utf8 で出力するため、このパッチにより eucjp を出力するようにします。パッチをあてたら、

```
perl hatena2list.pl > hatena
```

としてキーワードを取得します。その後、

```
perl mk_trie_regexp.pl hatena
```

とすると、先ほどのキーワードを読み込み「hatena.rx」に正規表現を出力してくれます。大体 1.6 Mbyte 程度になると思います。これを fswiki のログディレクトリ（標準では「./log」）に置いてください。

キーワード正規表現ファイルの設置に成功すると、hatenize プラグインが働きだし、そこかしこにリンクが作られると思います。この hatena.rx は、cgi からの書き込み権があると 3 日毎にはてなからキーワードを取得して自動更新します。ただ、先ほどのように、この処理は非常に重いため、自動更新をしても大丈夫か、検討する必要があるでしょう。書き込み権がなければ自動更新はしません。

■はてなキーワードリンクにスタイルを設定する

さて、このパッチによりあちこちにリンクが作られたと思います。しかし、このままだとあまりに目立ち、不自然すぎ（笑）そこで、これまで触れていなかった parser_add_style.patch の出番です。これは「lib/Wiki/Parser.pm」と「lib/Wiki/HTMLParser.pm」を変更し、はてなキーワードリンクにクラス属性を付与するパッチです。このパッチをあてて、ユーザ定義スタイルに、

```
a.hatena {
  text-decoration: none;
  color: #000000;
  border-bottom: dotted 1px #d0d0d0;
}
```

等を加えれば目立たなくなります。

それでは「どうぞご利用ください。」:-)

[追記] そういえば書いていなかった事が。当初 Wiki::InterWiki を乗っ取る予定でしたが、これって実は有効順位が高く、他ページへのリンクより優先されてしまう（笑）そんなわけで、Wiki::Keyword を乗っ取りました。

2005-9-17

Wiki::InterWiki を乗っ取るプラグイン完成

ちょっと遅くなりましたが、日記 /2005-9-11 で書いた Wiki::InterWiki の乗っ取りに成功しました。あとは実際にはてなキーワードを取得してリンクするプラグインを書くだけ。っていうかここからが本題ですね。