

Nucleus CMS 4.0の新機能

日本語版リリースマネージャー兼
メインコミッター
坂本貴史

Nucleus CMSとは？

- 2002年にベルギーで開発されたコンテンツマネジメントシステム
- その後オープンな開発体制となる
- 日本語版は有志が2003年に公開
- 現在の最新版は3.64
- 現在の開発版は4.0

Nucleus CMSの基本機能

- ウェブブラウザからのリクエストに応じて、整形したメッセージを出力する
 - HTML/XML/XHTMLなどのテキストを出力
- 出力するメッセージのひな形をDBMSやファイルシステムを利用して管理する
 - ひな形=スキン・テンプレート

Nucleus CMSのシステム要件1

- サーバー側
 - ネットワーク機能を持ったオペレーティングシステム
 - HTTPサーバー機能を持ったソフトウェア
 - Apache2/Nginxなど
 - DBMS (データベースマネジメントシステム)
 - MySQL/SQLite (実験段階) など
 - PHP5

Nucleus CMSのシステム要件2

- クライアント側
 - ネットワーク機能を持ったオペレーティングシステム
 - モダンなウェブブラウザ
 - Internet Explorer 8以上、Firefox、Operaなど

Nucleus CMSの制約

1. 文字符号化方式固定
2. マルチバイト文字列処理不能
3. 非ASCII文字を含むメールが受信後おかしい
4. 翻訳メッセージの部分的な適用
5. ロケール名の表現ルールがまちまち
6. RDMSとしてMySQLしか考慮していない
7. 多くの箇所でPHP4依存のコードが見られる

1. 文字符号化方式固定

- ISO-8859-1 固定
 - 表記文字のうち、英数字と一部のラテン文字のみ
 - 米国と西ヨーロッパ以外の国の表記文字を扱えない
 - どこもかしこもデフォルトのまま使ってるため
 - このままだと日本語などが扱えない
- UTF-8を使えない一部の携帯端末のために、入出力の文字符号化方式の変換機能が必要
 - 例えばShift-JIS

1の対策

- PHP内の処理をUTF-8に統一
 - 意図しない文字符号化方式の文字列の混入予防
- データベース内テーブルもUTF-8に統一
 - RDBMSがキャラクタを損失するのを予防
- (設定によりUTF-8以外も適用できる予定)
- 携帯端末とのメッセージ入出力部分に文字符号化方式の変換機能を追加

2.マルチバイト文字列処理不能

- ISO-8859シリーズとUTF-8の違い
 - 一文字を表すために1~4バイトが必要
- PHPの文字列処理のための関数は、シングルバイト文字列が基本実装
 - 代表例: `strlen()`, `substr()`
 - `strlen("あ")`は3を返す
 - `substr("あいうえお", 1, 2)`は不正な値を返す

2の対策

- PHPのiconv拡張の利用
 - POSIXでAPIが定義されている、文字符号化方式変換のための実装
 - PHPのiconv拡張はこのほかに、文字符号化方式を考慮した文字列処理関数を提供
 - 代表例: `iconv_strlen()`, `iconv_substr()`
 - `iconv_strlen("あ")`は1を返す
 - `substr("あいうえお", 1, 2)`は「いう」を返す
- Mbstring拡張はPHPのデフォルト実装ではない
 - グローバルにリリースするためPHPのデフォルトに合わせる

3. 非ASCII文字を含むメールが受信後おかしい1

- メール送信のためのルール (SMTPプロトコル) は、非ASCII文字を含むメールを考慮してなかった
 - RFC 5321: Simple Mail Transfer Protocol
 - RFC 5322: Internet Message Format
 - 正確にはサブジェクトなどのヘッダ情報に含まれる場合におかしくなる
- 後に、MIMEの仕組みでヘッダのエンコード可能に
 - RFC 2231: MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations

3. 非ASCII文字を含むメールが受信後おかしい2

- PHPのビルトイン関数mail()はヘッダをエンコードせずに送信する
- その結果、受信するとおかしいメッセージのメールになる
 - こんな感じ「Subject: =?iso-2022-jp?B?MjAwNS0wMi0xM4LMjluJypXxjZA=?=」

3の対策

- ヘッダのエンコードをしてあげればmail()でもちゃんと送信可能
- ヘッダのエンコードを行ってからmain()を実行する機能を追加

4. 翻訳メッセージの部分的な適用

- 管理画面と一部のテンプレートのみ
- スキン/テンプレート機能やプラグインオプションへの適用機能がない

4への対策

- NP_Textという、<%text%>に翻訳メッセージを適用するプラグインが本体に同梱されている
 - これをコア機能 (BaseActionsクラス) にマージ
- スキンディレクトリ内の翻訳ファイルを自動でインクルードする機能を追加
 - 「/skins/(ディレクトリ)/locales」ディレクトリに翻訳ファイルを配置しとく

5.ロケール名の表現ルールが不定 1

- 翻訳ファイル名をロケール名としている
 - Englishとかjapanese-utf8とか
- 中国語の翻訳ファイル名が変で内容が重複してる
 - 簡体中国語: chinese/simchinese
 - 繁体中国語: chineseb5/traditional_chines
- ロケール名の表現ルールが不定なため再利用しにくい
 - HTML要素のlanguage属性や、XML要素のxml:lang属性の値として使えない

5.ロケール名の表現ルールが不定 2

- 翻訳ファイルの文字符号化方式がファイル内に記入されている
 - 文字符号化方式を指定しての読込ができないので文字化けの原因に
- ルールを整えるなら、変わりにくく資料があるものが好ましい
 - 独自ルールだとドキュメントを用意する必要があってメンテナンスが面倒

5への対策

- RFCの援用
 - RFC5646: Tags for Identifying Languages
 - language-script-region
 - (言語)-(文字体系)-(地域)
 - それぞれのタグはIANAに登録されている
- 文字符号化方式をファイル名に付加
 - これもIANAに登録されている名前を利用
 - UTF-8とかWINDOWS-1250とか
- 結果「ja_Jpan_JP.UTF-8」のようなファイル名に

6.RDBMSとしてMySQLしか考慮 してない

- MySQL用のPHPビルトイン関数でしか動作テストしてない
 - sqlite用ビルトイン関数やPDO (PHP Database Object) の動作テストしてない
- MySQLの独自拡張SQL文を使っている

6の対策

- 下位互換性を維持しつつ、RDBMSの違いを吸収するクラスを作成
 - Dbクラスとその下位クラスとしてのPDO、MysqlPDOクラス、MysqlPDOStatementクラス
- コア機能が作成するSQL文の見直し
 - 順次作業中。Searchクラスがネック。
- SQLの標準仕様を採用
 - SQL89やSQL92に準拠予定

7. 多くの箇所でPHP4依存のコードが見られる

- PHP4サポートが3.5系まで続いた名残り
 - 3.61のリリースが2010/12/19
- 初リリース時のレガシーなコードが残ってる
- PHP5では使われない機能のコードが残ってる

7の対策

- コードのレビューと書き換え
 - すべてのコードに対して作業中。。。
- スキン/テンプレートで管理画面を構成可能にする
 - 表示周りのコードの再レビューと修正
- プラグインの基底クラスを抽象クラスに書換え
 - オーバーロードしなければならない/してはいけないメソッドを整理

Nucleus CMS 4.0の新機能

- インストーラーの改善
- スキン/管理画面/プラグインの多言語化
- 携帯端末向け文字符号化方式変換機能
- スキン/テンプレートで構成可能な管理画面
- スペシャルスキンパーツをスキンからインクルード可能に
- MySQL以外のRDBMSを利用可能に

Nucleus CMS 4.0いつリリース?

- 未定です
- コア機能をほとんど書き換えたメジャーバージョンアップになるので、十分なテスト運用をしてからの予定
 - アルファバージョンテスト(6月?)
 - ベータバージョンテスト(7月?)
 - リリースキャンディデートテスト(8月?)
 - リリース(9月?)
- 開発に参加してる人が3人
- みなさんもテストに参加してください!!