Ruby 初級者向けレッスン第 34 回

okkez @ Ruby 関西

2010 年 03 月 20 日

今回の内容

- るりまに貢献するいくつかの方法

今回のゴール

- るりまに貢献出来るようになる

るりまに貢献する方法

- ブログや twitter でるりまを使っている事を公言する
- ブログや twitter で要望や修正点を公開する
- Redmine や ML にパッチなしで修正点を投稿する
- Redmine や ML にパッチ付きで修正点を投稿する
- コミッタになって投稿されたパッチを適用する
- コミッタになってがんがん書く

ブログや twitter で〜

- 私が見つける可能性は低いので不具合報告はあんまり意味ないです
- @okkez に reply とかされていれば見るかもしれません

Redmine や ML に投稿する

- 登録方法は簡単です
- ML はメールを送信するだけ
  - http://redmine.ruby-lang.org/wiki/rurema/MailingList
- Redmine にチケットを起票するにはアカウントを登録すれば OK

コミッタになる

- okkez に「コミッタになりたいです」ってメールする
  - http://redmine.ruby-lang.org/wiki/rurema/SubversionRepository
- 流れてきたパッチを適用してコミットするだけの簡単なお仕事
- ライブラリのドキュメントを書く
- bitclust の改善

準備

なにはともあれソースが無いと始まりません。一式準備しましょう。

```
$ svn co http://jp.rubyist.net/svn/rurema/doctree/trunk rubydoc
$ svn co http://jp.rubyist.net/svn/rurema/bitclust/trunk bitclust
```

http://redmine.ruby-lang.org/wiki/rurema/SubversionRepository も参照してください。

bitclust/{bin,tools} にパスを通すと便利です。この後の説明では bitclust/bin にパスを通している前提で進めます。

次にデータベースを作成します。今のところ、エンコーディングは euc-jp 限定です。

```
$ bitclust.rb -d ./db-1.8.7 init version=1.8.7 encoding=euc-jp
$ bitclust.rb -d ./db-1.8.7 update --stdlibtree=src
```

二つ目のコマンドは小さいファイルを大量に作るので少し時間がかかります。rubydoc/refm/src/ にリファレンスのソースがあります。リファレンスの文法は RD ベースになっています。ただし、厳密にパースするために制限が増えています。

正確な情報については以下を参照してください。

- http://redmine.ruby-lang.org/wiki/rurema/ReferenceManualFormatDigest
- http://redmine.ruby-lang.org/wiki/rurema/ClassReferenceManualFormat
- http://redmine.ruby-lang.org/wiki/rurema/HowToWriteMethodEntry
- http://redmine.ruby-lang.org/wiki/rurema/FrequentlyAskedQuestions

rubydoc/ 配下の他のディレクトリに関しては以下のとおりです。

```
├── faq                 # 旧リファレンスの Ruby FAQ のデータ
└── refm                #
    ├── api             #
    │   └── src         # リファレンスのソース
    ├── capi            # C API のリファレンスのソース
    ├── doc             # トップページや「Ruby 言語仕様」など
    │   ├── news        #
```

```
│   ├── platform   #
│   └── spec       #
└── old            # 旧リファレンスの雑多なコンテンツのコピー
```

ついでに動作確認用の Ruby を各バージョン用意しておくといいと思います。

- http://redmine.ruby-lang.org/wiki/rurema/HowToInstallRubys

リンク先で紹介されているダウンロードサイトは回線が細かったと思うので無茶しないようにしてください。ファイルのダウンロードには各種ミラーサイトを使うようにしてください。
Windows の人は mswin32 版のバイナリを適当に使ってください。
Unix な人は以下のようにするといいと思います。

- http://gist.github.com/250796
  - 実際に okkez の環境で使っているスクリプト

自分の環境に合うように手直しして使うといいと思います。この方法で Ruby を用意しておくと forall-ruby で全てのバージョンの Ruby を一括実行することができます。バージョン間の動作の違いを調べるのに便利です。

修正方法

準備が出来たので修正方法を説明していきます。

簡単な修正

typo の修正などの簡単な修正は以下のようにします。

```
$ edit refm/api/src/_builtin/Array          # 編集して
$ LANG=C svn diff                           # 差分を確認して
$ bc-tohtml.rb refm/api/src/_builtin/Array \
  --target=Array#pop --ruby=1.9.2 > /tmp/a.html # preview する
$ LANG=C svn diff > foo.diff
```

簡単な修正の場合はこれで作成した差分を Redmine のるりまプロジェクトの Issue Tracker でチケットを起票してファイルを添付すれば ML にメールが流れます。LANG=C しているのは ja-JP.UTF-8 な環境では svn diff で作成される差分に UTF-8 な日本語が入るので文字コードが混在してちょっとイヤな気分になるのを防ぐためです。
コミッタの場合は、簡単な修正の場合はコミットしてしまっていいと思います。

大きめの修正

基本的な流れは同じですが、やっぱり大きな修正なのでしっかり確認します。

```
$ edit refm/api/src/_builtin/Array          # 編集して
$ LANG=C svn diff                           # 差分を確認して
$ cd refm/api
$ rm -rf ./db-1.8.7
$ bitclust.rb -d ./db-1.8.7 init version=1.8.7 encoding=euc-jp
$ bitclust.rb -d ./db-1.8.7 update --stdlibtree=src
$ cd -
$ LANG=C svn diff > foo.diff
```

こんな感じでデータベースを再構築してエラーが出ないことを確認すると良いです。あとは先ほどと同じように Issue Tracker でチケットを起票してファイルを添付すれば良いです。

コミッタの場合でも一度チケットを起票するのがいいかもしれません。しばらく反応が無ければコミットしてしまっていいでしょう。

TODOs

RubyKaigi2010 までにやることを列挙します。

- まだ揃っていないメソッドエントリを記述する
- レビュー依頼が出ているライブラリのドキュメントをレビューする
  - 文法が正しいかチェックする
  - 定められたルールどおりに記述されているかチェックする
  - 内容が正しいかチェックする
  - サンプルコードがちゃんと動くかチェックする
- 「Ruby 言語仕様」を記述する

まだ揃っていないメソッドエントリを記述する

コミッタの場合は以下のようにする。

- http://redmine.ruby-lang.org/wiki/rurema/Phase3WorkingScheme
- refm/api/ASSIGN ファイルを見て担当者が付いてなくて done になっていないライブラリを見つける
- ASSIGN ファイルに自分の svn アカウントを書いてコミットしてライブラリの担当者になる
- すべての #@todo タグを削除出来る状態にする
- コミットする

- ASSIGN の状態を done にしてコミットする

コミッタでは無い場合は以下のようにするといいでしょう。

- ASSIGN ファイルを見て空いてるライブラリを探す
- パッチを書いてチケットに添付する

レビュー依頼が出ているライブラリのドキュメントをレビューする

- ITS でカテゴリを ”doc:review” で絞ると一覧が出るのでそのなかから選んでレビューする
- OK であれば OK としてチケットにコメントを書く
  - 権限を持つ人が気付いたら適切なアクションをしてくれる
- NG であれば NG としてチケットにコメントを書く
  - コメントだけでも良いが、パッチがあるとなお良い
  - むしろ、パッチが無いと無視される可能性が高い

実際のレビューの様子

さて、実際のレビューの様子を以下に示します。

- まずは ITS をチェックしましょう
- abbrev が簡単そうなのでこれにしましょう
- refm/api/src/abbrev.rd をチェックします

  $ bc-classes abbrev 1.8.0 (2003-08-04): library not exist: abbrev

  ```
  180 181 182 183 184 185 186 187 187188dev191p420192dev
  ```

  Abbrev - o o o o o o o o o o o

1.8.1 以降で存在しているようです。

```
$ bc-methods -rabbrev --ruby=1.9.2 --diff=refm/api/src/abbrev.rd Abbrev
+Abbrev
+Abbrev#abbrev
```

出力の行頭に + が付いていますが module function なので問題ありません。よってこのライブラリのドキュメントはレビュー結果が OK となります。

メソッドエントリに不足がある場合や、内容が明らかにおかしい場合などはレビュー結果 NG となります。できるだけパッチを作成して元のチケットにコメントともに報告しましょう。

このチェックはコマンドの実行結果と実際のドキュメントを見て比較するだけなので簡単です。

各種ツールの詳しい使い方は wiki や ML の過去ログまたはヘルプを参照してください。bc-classes, bc-methods あたりをチェックすればいいでしょう。

Ruby 言語仕様を記述する

初級者には無理なので保留。。。

演習

- redmine.ruby-lang.org にアカウントをつくろう
- レビューしてみよう
- チケットにコメントしよう
- コミッタになろう

redmine.ruby-lang.org にアカウントをつくろう

まだアカウントを持っていない人は http://redmine.ruby-lang.org/ にアクセスしてアカウントを作ってください。ついでにるりまの ML にも登録しておこうと便利です。

レビューしてみよう

http://redmine.ruby-lang.org/projects/rurema/issues を見てレビュー出来そうなやつをレビューしてください。

チケットにコメントしよう

レビューしたら結果をチケットにコメントしてください。

るりまコミッタになろう

自信がある人はコミッタになるための手続きをしてみてください。

まとめ

- るりまの執筆は Ruby の勉強になるよ！
- るりまは人手不足なので貢献するチャンス！
- 簡単なお仕事から難しいお仕事まで幅広く活動できるよ！