PRTR - public database Generator

Kurzanleitung

ENDA GmbH & Co. KG -- Environmental Data Management Solutions

Berlin, 12. November 2014

Inhaltsverzeichnis

1 Kurze Programmbeschreibung

Dieses Programm extrahiert die für das Web-Portal 'http://www.thru.de/' relevanten Daten aus der e-PRTR Qualitätssicherungsdatenbank und erzeugt daraus eine performante SQLite Datenbank.

Dazu verbindet sich das Programm mit der in 'db_source.yml' angegebenen zentralen e-PRTR Datenbank, extrahiert die relevanten Daten und schreibt diese anschließend in die per 'db_target.yml' definierte Datenbank.

2 Voraussetzungen zur Inbetriebnahme

Folgende Voraussetzungen müssen für die Lauffähigkeit des Programms erfüllt sein:

- Ruby Version 1.9.1,
- Postgres Version 8.4,
- Ruby Postgres-Adapter pg Version 0.9.0
- Ruby SQLite-Adapter sqlite3 1.3.5 und sqlite3-ruby 1.3.3,
- Datenbank-Schnittstelle dbi und die dazugehörigen adapter dbd-pg 0.3.9 und dbd-sqlite3 1.2.5.

Abweichende Versionen können ebenfalls funktionieren.

3 Inbetriebnahme

Die folgenden Schritte beziehen sich darauf, dass Sie sich bereits im Hauptverzeichnis des Programms befinden.

3.1 Automatisierter Programmablauf

Bei dieser Version des Programmablaufs handelt es sich eigentlich um ein Wrapper-Script. Dieses umgibt und führt das in ruby geschriebene Hauptprogramm aus. Das Script wurde zur vereinfachten Bedienung programmiert. Nach wie vor können die ruby scripte von Hand ausgeführt werden, dazu mehr im Abschnitt '4.2 von Hand ausführen'.

Das Wrapper-Script hat den Name 'prtr2sqlite.sh' und kann aus dem Programm Hauptverzeichnis ausgeführt werden.

Eine Übersicht und Hilfe erhalten Sie mittels der Eingabe von './prtr2sqlite.sh -h'.

3.1.1 Setup

Falls das Setup bereits ausgeführt wurde, können Sie diesen Abschnitt überspringen.

3.1.1.1 Konfigurationsdatei für die Quelldatenbank anpassen

Passen Sie die Datei './setup/db_source.yml' so an, dass sich die benötigten Schnittstellendaten zur Verbindung mit der ePRTR Datenbank in der Datei befinden.

- Das Schema der Datei lautet:
  url: DBI:Pg:<db_name>:<host-url>
  user: <username>
  password: <password>

3.1.1.2 Konfigurationsdateien für die Zieldatenbanken anpassen

Die Konfigurationsdateien 'db_target_de.yml', 'db_target_en.yml' und 'db_target_inspire.yml' enthalten bereits die benötigten Schnittstellendaten, diese können oder müssen evtl. angepasst werden.
Achtung! Falls Sie die Dateien anpassen, sollten Sie auch die dazugehörigen Setup-Variablen in der Datei 'prtr2sqlite.sh' anpassen. Diese befinden sich in dem Abschnitt der mit 'setup' '/setup' markiert ist. Andernfalls kommt es bei der Ausführung des Scripts zu Fehlern.

3.1.1.3 Setup abschließen

Wechseln Sie in das Programm Hauptverzeichnis und führen Sie den Befehl ' ./prtr2sqlite.sh -n' aus.

Der Parameter '-n' bewirkt folgendes:

- Erstellen der Verzeichnisse 'data' und 'config'
- Kopieren der eben bearbeiteten '.yml' Dateien in das 'config' Verzeichnis
- Erstellen und Kopieren der benötigten SQLite Datenbank-Templates. Die Templates werden aus den im Verzeichnis 'create_sql' befindlichen '.sql' Dateien erzeugt und danach in die Verzeichnisse 'data' und '/dev/shm/' kopiert.

3.1.2 Programm ausführen

3.1.2.1 Vorbereitung

Zunächst sollten Sie überprüfen, ob sich im '/dev/shm/' Verzeichnis die benötigten Datenbankdateien befinden (prtr_de.db, prtr_en.db, prtr_inspire.db).
ls -la /dev/shm/

Ist dies nicht der Fall, kann einfach der Befehl './prtr2sqlite.sh -n' aus dem Programm Hauptverzeichnis ausgeführt werden.

3.1.2.2 Ausführen

Nun können die SQLite Datenbanken gefüllt werden. Eine Übersicht der Optionen erhalten Sie mittels './prtr2sqlite.sh -h'.

- Beschreiben der deutschen SQLite db
  ./prtr2sqlite.sh -d
- Beschreiben der englischen SQLite db
  ./prtr2sqlite.sh -e
- Beschreiben der deutschen inspire SQLite db
  ./prtr2sqlite.sh -i
- Beschreiben der deutschen, der deutschen inspire- und der englischen SQLite db
  ./prtr2sqlite.sh -a

3.1.2.3 Abschließen

Nachdem die Datenbanken mit den extrahierten Werten gefüllt wurden, werden sie automatisch in das Verzeichnis '/home/uba/prtr/db/' kopiert. Die Versionsnummer im Dateinamen der Datenbank (relXXX) wird automatisch inkrementiert.

ls -la /home/uba/prtr/db/

3.2 Von Hand ausführen

Wie bisher kann das eigentliche ruby Programm auch von Hand ausgeführt werden.
Es gelten dabei die gleichen Regeln wie bisher.

3.2.1 Setup

Falls das Setup bereits ausgeführt wurde, können Sie diesen Abschnitt überspringen.

3.2.1.1 Konfigurationsdateien anpassen

Im ersten Schritt müssen Sie die Konfigurationsdateien erstellen und anpassen.
Dazu wechseln Sie bitte in das Verzeichnis 'setup' und führen dort das Script 'setup.sh' aus.
Passen Sie dann die durch das Script erstellten Dateien im Verzeichnis 'config' an Ihre Datenbanken an.

- db_source.yml - benötigt Daten um sich mit der ePRTR Datenbank zu verbinden
- db_target_de.yml - benötigt Daten um sich mit der deutschen SQLite db zu verbinden
- db_target_en.yml - benötigt Daten um sich mit der englischen SQLite db zu verbinden
- db_target_inspire.yml - benötigt Daten um sich mit der deutschen inspire SQLite db zu verbinden
- Das Schema der Dateien lautet:
  url: DBI:Pg:<db_name>:<host-url>
  user: <username>
  password: <password>

Wir empfehlen beim Anpassen der Schnittstellendaten darauf zu achten, die SQLite Datenbanken in einen RAMfs-Mountpunkt zu legen, da sich ansonsten die Schreibzugriffszeiten der INSERT Befehle als Performance-Engpass erweisen können.

3.2.1.2 Datenbank-Templates erstellen

Erstellen Sie die benötigten SQLite Datenbanken und kopieren Sie diese in die von ihnen dafür vorgesehenen Verzeichnisse. Um die Datenbanken zu erstellen, können Sie die vorhandenen SQL-Statements aus dem Verzeichnis 'create_sql' verwenden. Eine einfache Möglichkeit dies zu tun wäre z.B. folgende:

*cat *_de.sql | sqlite3 prtr_de_template.db*

Dieser Befehl übergibt alle in '*_de.sql' enthaltenen Daten an sqlite3 und erzeugt daraus eine sqlite3 Datenbank mit dem Namen 'prtr_de_template.db'.

Der Befehlsteil '*_de.sql' steht hierbei für alle Dateien deren Namen mit '_de.sql' enden.

3.2.2 Programm ausführen

3.2.2.1 Vorbereitung

Bevor Sie das Programm ausführen, ist darauf zu achten, dass sich die SQLite Datenbanken bereits in den vorgesehenen Verzeichnissen befinden (wie in den Konfigurationsdateien angegebenen).

3.2.2.2 Ausführen

Wechseln Sie in das Hauptverzeichnis des Programms und führen Sie einen der folgenden Befehle auf:

- Beschreiben der deutschen SQLite db
  ruby lib/main.rb
- Beschreiben der englischen SQLite db
  ruby lib/main_en.rb
- Beschreiben der deutschen inspire SQLite db
  ruby lib/main.rb –inspire

3.2.2.3 Abschließen

Wichtig! Falls sich Ihre SQLite Datenbanken in einem RAMfs-Mountpunk befinden:

Kopieren Sie die erstellten und beschriebenen Datenbanken aus dem RAMfs-Mountpunk, da dieses Verzeichnis bei einem Neustart oder Ausschalten des Computers geleert wird.