



Open

In association with heise online

Search The H Open

Search

[Last 7 days](#)
[News Archive](#)
[Features](#)

« previous 1 2 3 next »

Reversible Migrations

Aaron Patterson (@tenderlove) made Rails 3.1 a lot smarter by implementing reversible migrations. Instead of using only self.up and self.down to manage your migrations, Patterson has included the #change method, which will allow Rails to "figure out" how to reverse a given migration on its own.

For example, consider the following Rails 3.0 standard migration, compared with one using #change:

```
# Rails 3.0
class CreateUsers < ActiveRecord::Migration
  def self.up
    create_table :users do |t|
      t.string :name
      t.string :phone
      # ... and so on ...
    end
  end

  def self.down
    drop_table :users
  end
end

# Rails 3.1
class CreateUsers < ActiveRecord::Migration
  def change
    create_table :users do |t|
      t.string :name
      t.string :phone
      # ... and so on ...
    end
  end
end
```

When running a rollback, Rails will "figure out" how to reverse the 3.1 version of the migration on its own. It simply applies the inverse of whatever was called for in the #change method.

Unfortunately, this particular improvement isn't perfect. In testing, some simple things, such as changing a column's definition after the table was created (to use an int(8) instead of an int(4) in PostgreSQL), couldn't be reversed because figuring out what the field should have been changed back to wasn't possible.

In this case, the standard #up and #down methods can be used. A minor difference from previous versions, the #up and #down methods for a migration are not scoped to the current class. In other words, simply use "up" and "down", not "self.up" and "self.down" for up and down migrations in Rails 3.1. For example:

```
class CreateUsers < ActiveRecord::Migration
  def up # no "self."
    create_table :users do |t|
      # ...
    end
  end

  def down # "no self."
    drop_table :users
  end
end
```

While most migrations of significant complexity will still need to use the standard #up and #down methods, for simple table definitions, the #change method for reversible migrations can be very useful.

Prepared Statements

New features of Rails 3.1



New in Rails 3.1
Reversible Migrations
has_secure_password

THE H OPEN HEADLINES

The H is closing down

Hardware Hacks: Fire, alarms, touchable boards and NFC rings

GitHub gets smart over open source licences

NSS 3.15.1 brings TLS 1.2 support to Firefox

Second Android signature attack disclosed

One month left for the EclipseCon Europe 2013 call for papers

THE H OPEN

Kernel Log: Coming in 3.10 (Part 4) - Drivers



Linux 3.10 will be able to use the video acceleration features offered by Radeon graphics cores. Systems with Intel graphics will wake from standby faster. Linux now has an input device driver for Apple's infrared receiver [more »](#)

The trouble with "Business Source"



The problem of creating funding in a new software business is a major one, and doubly so for open source based companies. Michael Widenius recently described his solution to the problem, "Business Source", claiming it delivers "most of the benefits of open source". The H took a look to see how that held up [more »](#)

Kernel Log: Coming in 3.10 (Part 3) - Infrastructure



Kernel developers have toned down an over-eager feature for protecting against the Samsung UEFI bug and added a function for reducing timer interrupt overhead. Improvements have also been made to Hyper-V support and instructions for reporting errors [more »](#)

Whatever happened to Google?

At RailsConf 2011, Aaron Patterson introduced [Prepared Statements for Rails 3.1](#).

Normally, Rails sends a standard SQL statement to the database, the database prepares a query plan for the statement, runs the statement, then returns it. This is a four-step process.

By using prepared statement caching, this process is reduced by half. In Rails 3.1, queries such as `SELECT * FROM table_name WHERE id = $1` will be sent. The database then caches the statement and returns a token to Rails. Later, when the query is actually needed, Rails sends back a token and a value for the missing element(s). This makes read operations much more efficient, because the statement is already cached and the query plan built in advance, meaning that the database has only to execute the query and return the result set.

In the above linked video, Patterson shows performance impact for three different databases: SQLite, PostgreSQL, and MySQL. This change increases performance in PostgreSQL and SQLite in all cases, and in MySQL in cases where complex queries are stored. Unfortunately, it actually degrades performance in MySQL with simple queries because MySQL does no advance query planning, meaning that storing the query is simply an extra step with no benefit. Additionally, executing prepared statements on MySQL requires exchanging network data twice for some reason.

Developers using MySQL as their production database have a difficult choice to make: to use prepared statement caching, or not. In cases of complex queries, it might be worth it. Because MySQL uses two network round trips when prepared statements are executed, the network latency between your database and application servers is a factor to consider. You would have to find out if total network latency, when factored in with the speed performance on said complex query (or queries), is still an overall improvement in speed. In some cases this may be true, but in others it won't be. On simple queries, MySQL's performance degrades when using prepared statement caching.

Engine Yard has a [blog post on benchmarks for SQL Server](#), stating that the performance is up to 10x faster for complex queries, and doubles speed on simple queries.

Patterson has implemented this new feature in such a way that developers don't have to do anything differently to benefit from prepared statement caching: the API is exactly the same, so these changes come "out of the box" for anyone building a Rails 3.1 application.

Role-based mass-assignment protection

Mass assignment protection has received an interesting upgrade with Rails 3.1. You can now specify roles on the model level with protected attributes. This is accomplished by allowing `attr_accessible` to accept a `:role` option, and `#update_attributes` an `:as` option. One would use this by defining what attributes are accessible by which roles, then specifying the role when `#update_attributes` is called.

Consider the following example:

```
# app/models/user.rb
class User < ActiveRecord::Base
  attr_accessible :username, :as => :admin
end

# app/controllers/users_controller.rb
class UsersController < ApplicationController
  def update
    @user = User.find(params[:id])
    @user.update_attributes(params[:user],
      :as => current_user.role.name.to_sym)
  end
end
```

In this example, we specify that the username attribute of a user can only be updated by an administrator. In the controller, responsible for triggering the update, we specify that the call to `#update_attributes` should include a symbol of the currently logged-in user's role. (Assume that `current_user` is the currently logged-in user, that it has `_one` role, and that the role has a "name" attribute, which we then convert to a symbol.)

Next: [has_secure_password](#)

« previous 1 2 3 next »

Print Version | Permalink: <http://h-online.com/-1285887>

Also on The H:

[Information leak in ZENworks Asset Management disclosed](#)
[Rails 3.1 now available](#)
[From Rails 2.3 to 3.0](#)
[Rails and Merb to merge for Rails 3](#)



Although Google continues to support a variety of open projects and people, Glyn Moody notes that, following recent changes to Google Code and Google Talk, concern is growing that something fundamental has changed [more »](#)



What's new in SUSE Linux Enterprise 11 SP3



Service Pack 3 includes numerous enhancements for virtualisation and, by adding Secure Boot support and new drivers, beefs up support for newer hardware. There are also numerous enhancements relating to server storage and networking [more »](#)

What's new in Fedora 19



In a nod to fans of classic desktop interfaces, the new Fedora includes a MATE variant and classic mode for GNOME. Systemd now takes care of containers and assigning network names. New drivers support 3D acceleration in newer Radeon graphics cards [more »](#)

What's new in Linux 3.10



A second SSD caching framework and support for the new Radeons' video decoder are two of the most important enhancements in Linux 3.10, which is now out. This version also includes several new and improved drivers and a change to the network stack to speed up HTTP connections [more »](#)

Free Software post-PRISM



The news has been full of talk of spying, whistleblowing and data mining. Glyn Moody looks at how open source has been used to threaten freedom and privacy and how it could be used to

"Classic" Rails Screencast updated
Ruby On Rails Security Guide published as free ebook



defend them [more »](#)

THE H SECURITY

Content Security Policy halts XSS in its tracks



Cross-site scripting (XSS) is one of the biggest problems faced by webmasters. The new Content Security Policy standard should finally provide some relief [more »](#)

Skype's ominous link checking: Facts and speculation



Our associate's discovery that URLs sent through Skype are then visited by Microsoft has caused quite a stir. A little more information has now emerged and leads to even more questions [more »](#)

Password protection for everyone



Those who heed well-intentioned recommendations and use a separate password for every service either require a photographic memory or the right techniques to keep the multitude of passwords under control [more »](#)

Two clicks for more privacy



"Like" buttons for Facebook, Google+ and Twitter present a privacy problem. A 2-click concept developed by heise online addresses this problem [more »](#)

THE H DEVELOPER

Java EE 7 at a glance



The next step for Java EE 6 was planned to be cloud support but the collapse of ambitious developer plans has meant Java EE 7 arrived with few fundamentally new aspects, representing more a consistent effort to round off existing features [more »](#)

Continuous database migration with Liquibase and Flyway



An application's version-controlled source code is stored in the repository. Why not that of the database? To reproduce arbitrary database states in development, test or production environments, two powerful Java libraries are at hand that can be seamlessly integrated into a build for an agile Continuous Delivery [more »](#)

Unit testing with Node.js



Consistent unit testing is a basic quality requirement in modern software development. Mocha is a framework for writing and executing such tests in Node.js [more »](#)

Ruby 2.0 - the 20th birthday present



On 24 February 2013, the Ruby community celebrated the 20th birthday of its programming language. Ruby 2.0, a new major release that includes various exciting new features, was released at the same time and The H looks at some of the major changes [more »](#)

HITS OF THE H

Linux Mint 15: A better Ubuntu for the desktop



The Linux Mint project has announced "the most ambitious release since the start of the project". Linux Mint 15 promises a focus on the desktop that Ubuntu has been neglecting lately. The H investigates whether the release delivers on these ambitions [more »](#)

What's new in Linux 3.9



The Linux kernel is finally able to use SSDs as hard-disk cache. Changes to the network subsystem promise to improve the way server jobs are distributed across multiple processor cores. Linux 3.9 also includes drivers for new AMD graphics chips and soon-expected Wi-Fi components from Intel [more »](#)

Replacing Google Reader



For a large number of internet users the current challenge is finding a replacement for Google Reader. The H's Fabian Scherschel has looked at the functionality that made Google Reader popular and the current best alternatives to the Reader experience [more »](#)

Attacking TrueCrypt



TrueCrypt is considered the software of choice for encrypting data. A small utility called TCHead systematically takes on this encryption [more »](#)

The H
Last 7 days
News Archive
Features

The H Open
Last 7 days
News Archive
Features

The H Security
Last 7 days
News Archive
Features

The H Developer
Last 7 days
News Archive
Features

The H Internet Toolkit
Update Check
Anti-Virus
Browsercheck
Emailcheck
Test SSL certificates
Whois query

My IP address
Traceroute
DNS query
Subnet calculator
MAC addresses

RFCs
Ping
Bandwidth calculator
Spam list query
IP addresses