

## Contents

- [Tips and Tricks](#)
- [Page pool](#)
- [Query cache](#)
- [Using gmail as a mail provider](#)
- [Adding contexts in Resin 3.1.9 \(without restart\)](#)
- [Hide the fact you are running Railo from users](#)
- [Getting the column list of a query object, case-sensitive!](#)
- [Use a custom function for the cfdirectory "filter" attribute](#)
- [URL Rewriting](#)
- [Other Tips](#)

## Tips and Tricks

If you have any tips that you would like to share, sent them to [contact@getrailo.org](mailto:contact@getrailo.org)

### Page pool

When you call a template in Railo, the template lands in the page pool for reuse. The pages in the page pool are refreshed depending on the settings you have made in the Performance/Caching area of the Railo administrator. Now if you want to make use of the setting "never" which will never look for changed templates during the lifetime of a server instance, you should use the built in function `pagePoolClear()` in order to flush the template cache (page pool).

If you want to see what pages are inside the page pool you can use the built in function `pagePoolList()`. The function returns an array with all the pages in the page pool

### Query cache

If you are using the query cache and you would like to flush the cache you can always use the tag in order to do so. The problem hereby is that this call flushes the complete cache which is a waste. In Railo you can use some additional attributes like `filter`, `filterignorecase` and `result`.

The attributes `filer` and `filterignorecase` allow you to flush objects from the cache that match the filter. The filter gets applied like the sql statement like. So if you use the following tag:

```
action="clear" filter="susi">
```

Then all elements in the query cache that are containing the string `susi` are flushed from the cache. This means you can easily remove all queries for a certain table if a table is updated on the

database. The attribute `filterignorecase` does the same without obeying the case.

The next new attribute Railo has introduced is `result` which can be used with the new action size. If you execute the following tag:

```
action="size" result="cachesize">
```

the variable `cachesize` will contain the number of elements in the cache.

## Using gmail as a mail provider

```
smtp.gmail.com (use authentication)
Use Authentication: Yes
Use STARTTLS: Yes (some clients call this SSL)
Port: 465 or 587
```

Additional resource: <http://jamiokruger.com/blog/index.cfm/2009/2/13/cfmail-using-Gmail-SMTP>

## Adding contexts in Resin 3.1.9 (without restart)

if you create a directory inside the Resin root called `hosts` and you create different directories in there named like the host name you expect, you just need to place a `host.xml` file in there that contains the following:

```
xmlns="http://caucho.com/ns/resin">
  www.client.com
  id="/" root-directory="D:/sites/someclient"/>
```

And then the context is available at runtime without having to restart the engine. So if you add a new website to the server then you just need to complete the above steps and then you can access the site.

So my Resin directory looks like this:

- resin
  - hosts
    - Client1
      - host.xml
    - Client2
      - host.xml

etc.

## Hide the fact you are running Railo from users

Sometimes you want to hide from users of your site, what underlying technology you are using. This takes a few steps:

1. Go to railo admin settings/error and replace error template with a custom error template that is plain or looks like an ASP or PHP error page, or of course, whatever template you want.
2. Add `<!-- CFMLServlet -->` at the top of this template.
3. Go to railo admin settings/output and remove flag "Output Railo version"
4. Go to WEB-INF/web.xml or etc/webdefault.xml or config/appdefault.xml depending on which servlet container you are running
5. Change all existing servlet-mappings from servlet "CFMLServlet" as follows
6. Replace `cfm`, with `asp` and `cfc` with `php` or whatever you want. e.g:

```
CFMLServlet
*.asp
```

7. Go to existing application.cfc or create one, if there is more than one, do this for every application.cfc or extend a cfc in every application.cfc with the following function:

```
name="onRequest" output="yes">
  name="path">
    var local.ext=mid(arguments.path,len(arguments.path)-2,3)>
    <!-- extension asp is used for cfm calls --->
    local.ext EQ "asp">
      var local.pathmod=mid(arguments.path,1,len(arguments.path)-3)&"cfm">
    <!-- extension php is used for cfc calls --->
    local.ext EQ "php">
      var local.pathmod=mid(arguments.path,1,len(arguments.path)-3)&"cfc">

    local.pathmod=arguments.path>

  showdebugoutput="no">
  template="#local.pathmod#">
  true>
```

8. Restart railo

## Getting the column list of a query object, case-sensitive!

It's really simple:

```
caseSensitiveColumnList = queryObject.getColumnlist(false) />
```

Note: if you typed the actual column names in your SELECT statement, like in "SELECT

userID, userName from users", then the exact column names you used there will be returned. But if you use "SELECT \* from users", then the actual table column names are returned. See [this blog post](#) for more info.

## Use a custom function for the cfdirectory "filter" attribute

Read more about this in the [Railo CFDirectory documentation](#).

## URL Rewriting

- See [Installation:URLRewriting](#)

## Other Tips

- [Implicit Conversions](#)
- [Magic Functions](#)
- [Serialize all kinds of data by using serialize\(data\)](#)
- [Setting up secure LDAP](#)