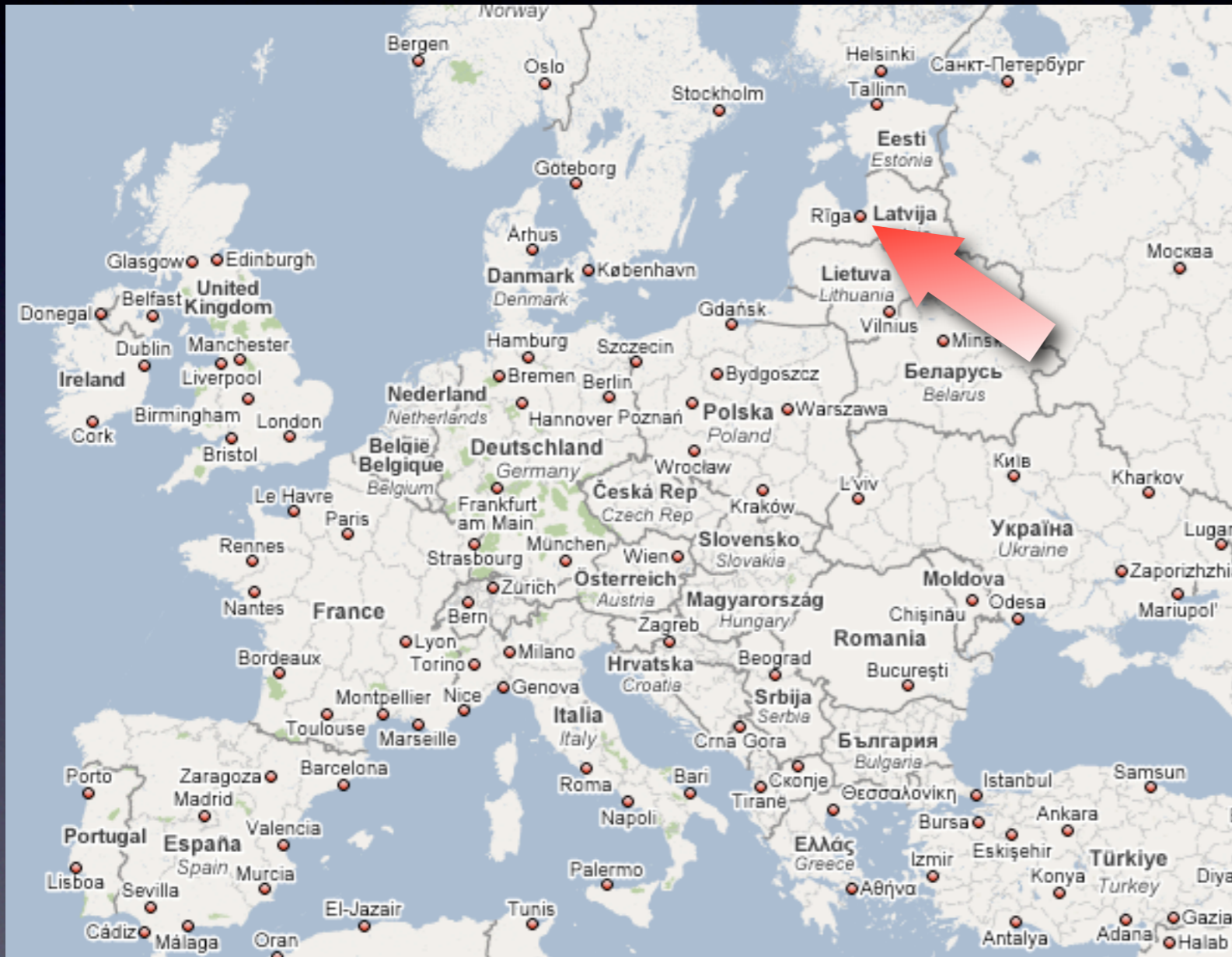


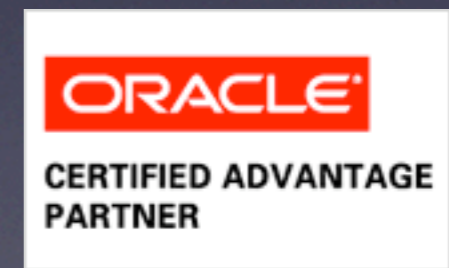
Using  Ruby with 

Where I am from?



Raimonds
Simanovskis

Riga, Latvia



Ruby => Oracle

Ruby application



ruby-oci8



Oracle Instant
Client



SQL*Net

Oracle Database

```
require 'oci8'  
OCI8.new('scott', 'tiger').exec(  
  'select * from emp') do |r|  
  puts r.join(',')  
end
```

Oracle and Mac :(

ORACLE Mix ^{Beta} New Features! Logged in as Raim

Home People Groups Ideas Q&A

Latest Ideas Greatest Ideas My Favorite My Ideas from everyone

135 Want it **We need the Oracle clients (OCI, JDBC) for the Apple Intel OSX platform**
Proposed by Dieter Oberkofler 4 months ago

After having waited for about 10 years we finally need a real support of the Apple platform including the current Intel OSX platform. OCI, JDBC are a must. A complete port of the database should follow soon.

Remove from favorites 36 comments Tags: apple, osx, intel, oci Products: Database

103 Want it **Compile Oracle XE for Intel OS X**
Proposed by Toby Tripp 4 months ago

Encourage Ruby and JRuby development on the Oracle stack by making a client available for the most popular development platform in the Ruby community: Mac OS X.

Remove from favorites 35 comments Tags: osx, mac, database, portability, developers, ruby, rails Products: Database and Express Edition

Issues with Mac and Oracle Instant Client

- No Oracle Instant Client for Intel Mac
 - Workaround – run Ruby in PowerPC mode
- Good news – Intel Mac version is arriving soon
 - Beta version working OK
 - Only on Leopard

JRuby => Oracle

Ruby application



JRuby



JDBC driver



SQL*Net

Oracle Database

```
require "java"
sql =
  JavaSQL::DriverManager.getConnection(db, user, password)
statement = sql.createStatement
status = statement.execute
"select * from parts;"
rs = statement.getResultSet()
```

Ruby on Rails 2.0 => Oracle

```
gem install activerecord-oracle-adapter \  
  --source http://gems.rubyonrails.org
```

```
# additional adapter patching needed - search in blogs...
```

Ruby on Rails 2.0 => Oracle

database.yml

development:

adapter: oracle

database: XE

username: blogdemo

password: blogdemo

encoding: utf8

cursor_sharing: similar

Employee.find(1)



ActiveRecord

```
SELECT * FROM employees  
WHERE (employees.id = 1)
```



Oracle optimizer

```
SELECT * FROM employees WHERE  
(employees.id = :SYS_B_0)
```

Oracle Data Types

Ruby	Rails	Oracle
Fixnum	:integer	NUMBER
Float	:float	NUMBER
BigDecimal	:decimal	NUMBER, DECIMAL
Time	:datetime	DATE
Time	:time	DATE
Date	:date	DATE
String	:string	VARCHAR2
String	:text	CLOB
True/FalseClass	:boolean	NUMBER(1), CHAR(1)

Calling PL/SQL from Ruby

```
require "oci8"
conn = OCI8.new("hr", "hr", "xe")

cursor = conn.parse <<-EOS
BEGIN
    :return := test_uppercase(:p_string);
END;
EOS
cursor.bind_param(':p_string', "xxx", String)
cursor.bind_param(':return', nil, String, 4000)
cursor.exec
puts cursor[:return]
cursor.close
```

ruby-plsql

```
gem install ruby-plsql
```

```
require "ruby-plsql"  
plsql.connection = OCI8.new("hr", "hr", "xe")  
  
puts plsql.test_uppercase('xxx')
```

ruby-plsql

```
plsql.connection = OCI8.new("hr", "hr", "xe")

plsql.test_uppercase('xxx') # => "XXX"
plsql.test_uppercase(:p_string => 'xxx') # => "XXX"
plsql.test_copy("abc", nil, nil) # => { :p_to => "abc",
# :p_to_double => "abcabc" }
plsql.test_copy(:p_from => "abc", :p_to => nil, :p_to_double => nil)
# => { :p_to => "abc",
# :p_to_double => "abcabc" }
plsql.hr.test_uppercase('xxx') # => "XXX"
plsql.test_package.test_uppercase('xxx') # => 'XXX'
plsql.hr.test_package.test_uppercase('xxx') # => 'XXX'

plsql.logoff
```

Thank You!

<http://blog.rayapps.com>