Rails Adapter/Driver for IBM Data Servers

Introduction

IBM_DB, the Rails Adapter and Ruby Driver for IBM Data Servers, supports development of web-applications using Ruby on Rails (ROR) Framework with IBM Data Servers. The ROR framework is officially tested and supported on DB2 data servers and Informix Dynamic Server (IDS) 11.10 and above.

To begin with, one can start using the no-charge version of DB2, available to the community. Currently DB2 is the only database system with native XML support. To download the free to develop, deploy, and distribute data server, which has no size, time, or user limit, with the following features

- Highly resilient and proven technology
- Seamless integration of XML and relational data
- Optimizes Web 2.0 and SOA applications

Visit http://www-01.ibm.com/software/data/db2/express/download.html

Getting started with IBM Data Servers on Rails

IBM_DB can be used to setup ROR environment with both DB2 and IDS. To get started with, download the free version of DB2 and start developing the ROR applications using DB2. Below is the step-by-step procedure to setup the environment and get going with IBM_DB with Rails.

Setting up IBM Data Server and Rails Environment

1) Download and install DB2 or IDS from the site http://www-306.ibm.com/software/data/servers/. (If setting up the environment to connect to a IBM Dataserver on a remote machine refer to section "*Setting up the environment for Remote conections*")

2) Download and install the latest version of ruby from http://www.ruby-lang.org/en/downloads/.

3) Install the rails gem and its dependencies by issuing the command

```
gem install rails --include-dependencies
```

Note: - DB2 version 9 FixPack 2 or DB2 8 FixPack 15 and above is required.

To connect to an IBM Data server (DB2 on LUW/ DB2 on IBMi/ DB2 on zOS) you will need the IBM Call Level Interface (CLI), a callable SQL interface to IBM data servers that is Open Database Connectivity (ODBC) compliant. The IBM_DB gem contains two components namely, the IBM_DB adapter (Rails Adapter for IBM DataServers) and the IBM_DB driver (ruby driver for IBM Dataservers). IBM_DB adapter has a direct dependency on the IBM_DB driver that uses IBM Dataserver Driver for ODBC and CLI to connect to IBM data servers. Hence you will need to ensure that CLI in installed and accessible in your environment. If you have the DB2 server installed on your system then CLI should be automatically available, else you will need to download the Client package separately and install them.

Setting up the environment for Remote connections

To connect to a remote DB2 server, at minimum you will need to install the IBM Data Server Driver package (DS Driver) http://www-947.ibm.com/support/entry/portal/Overview/Software/Information_Management/IBM_Data_Server_Client_Packages, which is a small footprint package of IBM Data Server Client Packages that contains the ODBC compliant CLI driver. Installing it is as simple as just extracting the downloaded archive on a Linux based system and setting the environment variable **LD_LIBRARY_PATH** to the **lib** directory under the extracted folder. On windows run the **.exe** file downloaded and follow the instructions. With this the client environment is ready to access the remote DataServer from your Ruby/ Rails application.

Note: You will not need the complete DB2 installed on your client machine to connect to DB2 on a remote machine. Just the above mentioned IBM Dataserver Driver package will be sufficient, unless you need to perform other database operations like Create a database, Catalog a remote database, Dataserver administration etc.

Installing the IBM_DB Adapter and Driver as a Ruby gem

Ruby Gems is the standard packaging and installation framework for libraries and applications in the Ruby runtime environment. A single file for each bundle is called a gem, which complies with the package format. This package is then distributed and stored in a central repository, allowing simultaneous deployment of multiple versions of the same library or application. Similar to package management and bundles (.rpm, .deb) used in Linux distributions, these gems can also be queried, installed, un-installed and manipulated through the gem end-user utility.

To enable ruby runtime to access IBM Data Servers, follow the steps below to install the IBM_DB gem on respective platforms

1. **Linux/Unix/MacOS X Platforms: -**

Step 1: - Set the Environment variables *IBM_DB_INCLUDE* and *IBM_DB_LIB* as below

$ **export IBM_DB_INCLUDE=DB2HOME/include**

$ **export IBM_DB_LIB=DB2HOME/lib**

Where *DB2HOME* is the *sqllib* directory under the DB2 instance's home or directory where the IBM Data server Driver (DS Driver) is installed.

Eg.

```
$ export IBM_DB_INCLUDE=/home/db2inst1/sqllib/include
$ export IBM_DB_LIB=/home/db2inst1/sqllib/lib
```

Note: -

1) If installing ibm_db version below 1.0.1 then set IBM_DB_DIR, instead of IBM_DB_INCLUDE, to DB2HOME

Eg. $ *export IBM_DB_DIR=/home/db2inst1/sqllib*

2) Setting of IBM_DB_LIB is to be done appropriately according to the architecture for which Ruby is compiled. If Ruby is 32-bit then set IBM_DB_LIB to directory ***lib32*** under DB2HOME. If Ruby is 64-bit then set IBM_DB_LIB to directory ***lib64*** under DB2HOME.

3) The ***lib*** directory under DB2HOME is a link to either ***lib32*** or ***lib64***. Depending on the Architecture for which DB2 is been installed, you can also set the path to lib directory accordingly.

Step 2: - Source the DB2 profile (optional)

```
$ source /home/db2inst1/sqllib/db2profile
```

Step 3: - Issue the command *gem install ibm_db*

```
$ gem install ibm_db
  Building native extensions. This could take a while...
  Successfully installed ibm_db-2.0.0
  Installing ri documentation for ibm_db-2.0.0...
  Installing RDoc documentation for ibm_db-2.0.0...
```

2) **Windows: -**

Step 1: - Issue the command *gem install ibm_db*

```
C:\>gem install ibm_db
   Successfully installed ibm_db-2.0.0-mswin32
   Installing ri documentation for ibm_db-2.0.0-mswin32...
   Installing RDoc documentation for ibm_db-2.0.0-mswin32...
```

With this the ibm_db gem is successfully installed and the functionality is immediately accessible from any ruby script (therefore application) in the Ruby runtime environment, through:

```
require 'rubygems'
require 'ibm_db'   #(In windows environment this will be require 'mswin32/ibm_db')
```

Note:

1) On Windows if using ruby compiled with mingw32 or ruby 1.9 (mswin32 or mingw32) you will need to manually download the gem from http://rubyforge.org/frs/?group_id=2361&release_id=38242 to your local drive, for the respective version of ruby, and install the gem from the download location. IBM_DB gem for ruby version 1.8 is named *ibm_db-2.0.0-mswin32.gem* and for ruby version 1.9 it is named *ibm_db-2.0.0-mswin32-rb191.gem*

E.g.: Say you downloaded the gem for ruby-1.9 to *C:\gem_downloads* then issue the install command from this location, as below

```
C:\ gem_downloads>gem install ibm_db-2.0.0-mswin32-rb191.gem
```

Installation verification

Follow the commands as below to verify the installation. This process can be followed to verify installation against DB2 server on LUW, DB2 on i5 and DB2 on zOS and IDS. For details on necessary tools required etc for connecting to IBMi and DB2 on zOS refer to section "*Accessing DB2 on zOS and DB2 on IBMi*".

```
C:\>irb
```

```
 irb(main):001:0> require 'mswin32/ibm_db'  (if using Linux
based platform then issue require 'ibm_db')
=>true
irb(main):002:0> conn = IBM_DB.connect
'devdb','username','password'
=>#<IBM_DB::Connection:0x2dddf40>
#Here 'devdb' is the database cataloged in client's
#database directory.
#To connect to a remote database you
#will need to specify all the #necessary attributes like
#hostname, port etc as follows.
#IBM_DB.connect('DRIVER={IBM            DB2            ODBC
DRIVER};DATABASE=devdb;HOSTNAME=myhost;PORT=60000;PROTOCOL
=TCPIP;UID=uname;PWD=secret;','','')
irb(main):003:0>  stmt  =  IBM_DB.exec  conn,'select  *  from
cars'
=> #<IBM_DB::Statement:0x2beaabc>
irb(main):004:0> IBM_DB.fetch_assoc stmt
#This will fetch the first row from the result set
irb(main):005:0> IBM_DB.close conn # Close the connection
irb(main):006:0> quit
```

Accessing DB2 on zOS and DB2 on IBMi

To access DB2 on IBMi and DB2 on zOS Data Servers you will need to install the *DB2 Connect* [http://www-01.ibm.com/software/data/db2/db2connect/] license along with the DS Driver Package (mentioned in the sections above). Refer to http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.apdv.cli.doc/doc/r0024162.html for more information on Licensing and configuring the DS Driver for connecting to DB2 on IBMi and DB2 on zOS.

Building a Rails Application to access IBM Data Servers

Follow the procedure below to develop a small Rails application that will access IBM Data Server using the IBM_DB gem.

Step 1: - create a new Rails application by issuing the following command

```
C:\>rails newapp
create
create  app/controllers
create  app/helpers
```

```
create  app/models
create  app/views/layouts
create  config/environments
create  config/initializers
create  db
[……]
create  log/server.log
create  log/production.log
create  log/development.log
create  log/test.log
```

Step 2: - Get into the newly created directory (**newapp**)

```
C:\>cd newapp
C:\newapp>
```

Step 3: - Edit the database.yml file with the appropriate data for the development section. A snippet showing the development section in the *database.yml* file is shown in *Listing 1* below

```
development:
  adapter:  ibm_db
  username: db2inst1
  password: secret
  database: devdb # Database name
  #schema:  db2inst1
  #host:    localhost #Host on which the database resides
  #port:   50000  #port  to  which  the  remote  Dataserver  is
  listening
  #account:     my_account
  #app_user:    my_app_user
  #application: my_application
  #workstation: my_workstation
  parameterized: false
```

Listing 1: Snippet showing development section of database.yml file

See section "*Configuring Rails application connections to IBM Data Servers"* for more information on the parameters in the *database.yml* file

Note:

1) If using Rails version less than 2.0 then you will need to register the ibm_db adapter to the list of connection adapters in Rails framework by manually adding ibm_db into <RubyHome>\gems\1.8\gems\activerecord-

1.15.6\lib\active_record.rb, approx around line 77, in the list of Rails Connections Adapter

…

RAILS_CONNECTION_ADAPTERS = %w(mysql postgresql sqlite firebird sqlserver db2 oracle sybase openbase frontbase ibm_db)

…

Step 4: - Create a model and scaffold as below

```
C:\>ruby script/generate scaffold Tool name:string serial:integer
      exists  app/models/
      exists  app/controllers/
      [….]
      create    db/migrate
      create    db/migrate/20080716103959_create_tools.rb
```

Step 5: - Issue the Rails migrate command to have the table (*tools*) created in the database (*devdb*)

```
      C:\ >rake db:migrate
      (in C:/ruby trials/newapp)
==        20080716111617       CreateTools:       migrating
======================================
     -- create_table(:tools)
      -> 0.5320s
==    20080716111617    CreateTools:    migrated    (0.5320s)
============================
```

With this you have the table *Tools* created, in database *devdb***,** and the Rails app is ready to be accessed and perform the necessary operations on the table Tools.

Step 6: - Issue the following command to test the application

```
C:\ruby trials\newapp>ruby script/console
      Loading development environment (Rails 2.1.0)
      >> tool = Tool.new
      =>  #<Tool  id:  nil,  name:  nil,  model_num:  nil,
 created_at: nil, updated_at: nil>
      >> tool.name = 'chistel'
      => "chistel"
      >> tool.serial = '007'
      => "007"
```

```
>> tool.save
=> true
>> Tool.find :all
=> [#<Tool id: 100, name: "chistel", model_num: 7,
created_at: "2008-07-16 11:29:35", updated_at: "2008-07-16
11:29:35">]
>>quit
```

Configuring Rails application connections to IBM Data Servers

To configure host data server connections for a Rails application, edit the database configuration file *database.yml* under rails_application_path>\config\ by entering the following connection attributes:

```
# == Mandatory parameters
adapter:    'ibm_db'       // IBM_DB Adapter name
username:   'db2user'      // data server (database) user
password:   'secret'       // data server (database) password
database:   'DEVDB'        // database name (remote or catalog entry alias)
# == Optional (highly recommended for data server auditing and monitoring
purposes)
#  schema:     'rails123'     // name space qualifier
#  account:    'tester'       // OS account (client workstation)
#  app_user:   'test11'       // authenticated application user
#  application: 'rtests'      // application name
#  workstation: 'plato'       // client workstation name
# == Remote TCP/IP connection (required when no local database catalog
entry available)
#  host:       'Socrates'     // fully qualified hostname or IP address
#  port:       '50000'        // data server TCP/IP port number
```

security: 'SSL' // optional parameter enabling SSL encryption available only from DB2 version V95fp2 and above

authentication: 'SERVER' // AUTHENTICATION type which the client uses to connect to the database server. By default value is SERVER

timeout: 10 // Specifies the time in seconds (0 - 32767) to wait for a reply from server when trying to establish a connection before generating a timeout

== **Parameterized Queries Support**

parameterized: false // Specifies if the Parameterized Queries support of the IBM_DB Adapter is to be turned on or off

Note: -

1) When schema is not specified, the username value is used instead.
2) The default setting of parameterized is false
3) Changes to the database connection configuration are only relevant when the Rails environment is initialized during server startup. Any changes after the fact is not going to affect the connections that are created.
4) Schema, account, app_user, application and workstation are not supported for IDS.

IBM_DB and JRuby

IBM_DB is not supported on JRuby. This is because (as stated in the JRuby Wiki [http://kenai.com/projects/jruby/pages/GettingStarted_-_Installing_and_Using_Ruby_Gems], "Basics of Getting JRuby Running"), " Many Gems will work fine in JRuby, however some Gems build native C libraries as part of their install process. These Gems will not work in JRuby unless the Gem has also provided a Java equivalent to the native library."

The IBM_DB adapter relies on IBM_DB driver (C extension) and the IBM Driver for ODBC and CLI to access databases on IBM data servers. Hence IBM_DB cannot be used with Jruby for accessing the IBM Dataservers. One can use JDBC_adapter to access DB2 when on Jruby.

Heap size issues with DB2 on Rails

Rails applications on DB2 9 requires the *APPLHEAPSZ*, database configuration parameter, to be set to 1024. You will need to set this parameter for each database for which you will be running DB2 on Rails applications. Following is the command syntax for updating the *applheapsz* parameter:

```
db2 update db cfg for <database_name> using APPLHEAPSZ 1024
```

To enable this parameter, you need to restart your database.