

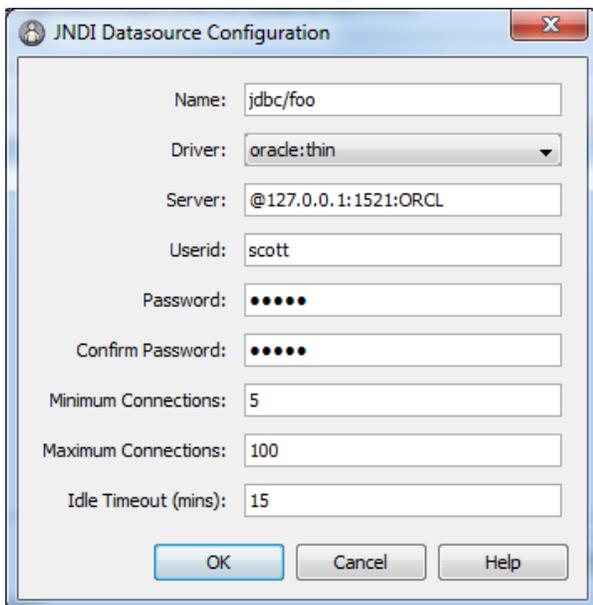
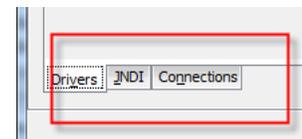
# An Option for Including External Database Content in a Template

## Objective

It is often desirable to include data from an External DataSource in a Template being published by the CMS. This document should provide some background to using the `$rx.db.get` JEXL function in a Velocity Template.

## DataSource configuration

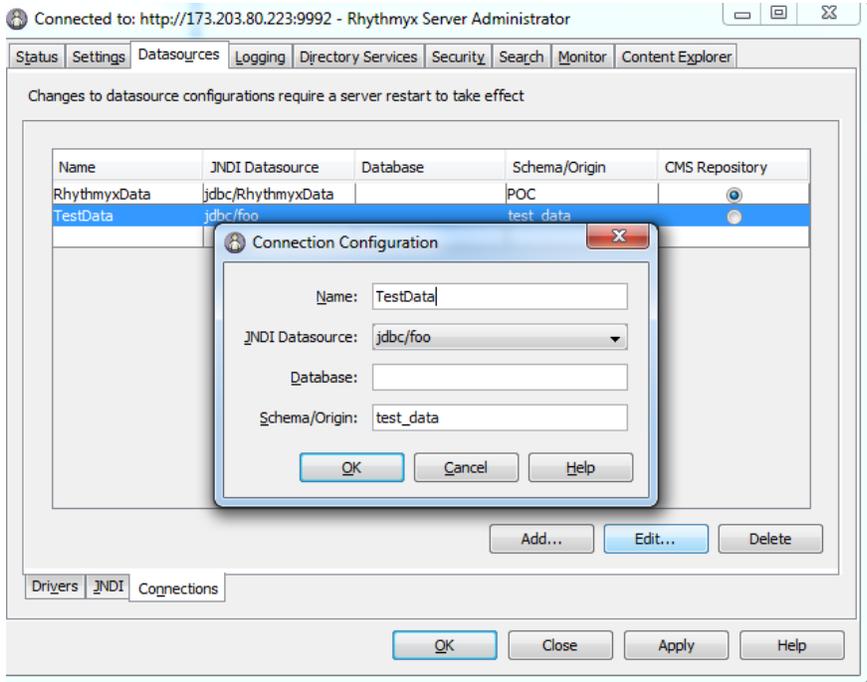
First you'll need to create a DataSource and Connection entry in the Server Administrator. On the "Datasources" tab of the Rhythmyx Server Administrator you'll see two Sub-Tabs, JNDI and Connections.



Once you've located the JNDI tab, you can create a new DataSource Configuration which should point to your External Database. In the example shown to the left, I have created a DataSource Config named "jdbc/foo" which is connecting to an Oracle server on the local system.

It should be noted that the DataSource Configuration only references a Server and the User to log in with. In

After successfully creating the DataSource Configuration, we can move to the Connections Tab. You'll notice that there is already a DataSource created here for the CMS Repository. You'll want to add a new Connection which references our previously created DataSource, as well as the Database/Origin (for MSSql) or Schema (for Oracle) .

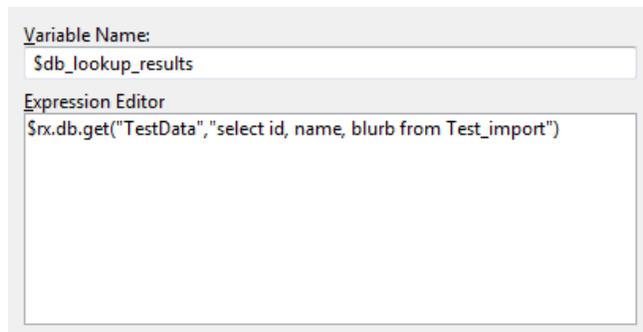


After these two Configurations are created, Apply your changes in the server and re-start the CMS to have the changes picked-up.

## Including Data in a Template

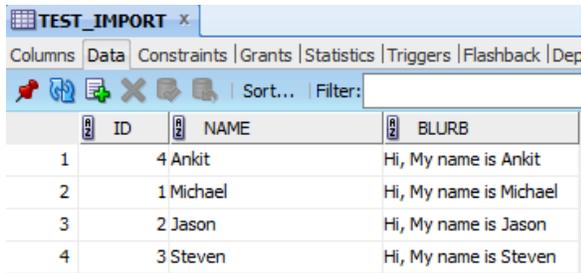
After the CMS has been re-started, we can log into the Workbench and utilize the JNDI Configurations to include content in a Template. We can use the `$rx.db.get` JEXL function to return the results of a SQL Query. This JEXL function takes two arguments, the Datasource Name and the SQL Statement to run.

```
$rx.db.get (datasource,sqlselect)
```



You can use this function in a Binding which will return a list of results. In the example shown here, we'll run the SQL Statement "select id, name, blurb from Test\_Import" against the TestData Connection we created previously.

Here is the Table Data that I've used in this example:



ID	NAME	BLURB
4	Ankit	Hi, My name is Ankit
1	Michael	Hi, My name is Michael
2	Jason	Hi, My name is Jason
3	Steven	Hi, My name is Steven

The `$db_lookup_results` variable will be populated with a java Array List we can loop through. For instance, the following lines of code will loop through the resultant array and create a simple table with a row for each item returned from the query.

```
<table>
  #foreach ($row in $db_lookup_results)
    <tr>
      <td>$row.ID</td>
      <td>$row.NAME</td>
      <td>$row.BLURB</td>
    </tr>
  #end
</table>
```

Resulting in the following output in the assembled item:

```
<table>
  <tr>
    <td>4</td>
    <td>Ankit</td>
    <td>Hi, My name is Ankit</td>
  </tr>
  <tr>
    <td>1</td>
    <td>Michael</td>
    <td>Hi, My name is Michael</td>
  </tr>
  <tr>
    <td>2</td>
    <td>Jason</td>
    <td>Hi, My name is Jason</td>
  </tr>
  <tr>
    <td>3</td>
    <td>Steven</td>
    <td>Hi, My name is Steven</td>
  </tr>
</table>
```