

# DBQuery (v1.1)

## 1. Description

Get data from any JDBC capable database and show resultset in IRC channel.

Using SELECT statements towards any JDBC capable database, you can get resultsets returned to your IRC channel. The complexity of the SELECT statements is only limited by the functionality of the loaded JDBC driver.

## 2. Configuration

The configuration file for this plugin is **system/DBQuery/DBQueryConfig.xml**

### 2.1. datasource

First of all you need to specify a datasource, in other words: provide connectivity information to your database. That means telling the plugin which database to use, which driver to load and which username and password needed to connect to the database.

```
<datasource>
  <driver>com.mysql.jdbc.Driver</driver>
  <jdbcurl>jdbc:mysql://localhost/cccp</jdbcurl>
  <username>cccp</username>
  <password>cccp</password>
</datasource>
```

Please take a look at system/DBQuery/readme.txt for datasource samples for other databases, and to find info on where to download a JDBC driver for your database of choice. By default DBQuery is only shipped with JDBC drivers for MySQL and PostgreSQL. If you are using another database vendor, you need to look at system/DBQuery/readme.txt to find out where to download a JDBC driver for your database. Save the downloaded JDBC driver's JAR file in the system/DBQuery directory.

### 2.2. commandShowTriggers

A command to show which SELECT statements are in the current configuration. The output attribute isn't actually used for this command. It is just provided for compatibility, so don't mess with it. The result is allways returned as DCC chat.

- **authLevel** - ANY, ADMIN, MASTER, TRUSTED, NONE
- **output** - PM

```
<commandShowTriggers authLevel="none" output="pm">dbq</commandShowTriggers>
```

### 2.3. columnSeparator

A character or a series of characters to use as separator between columns in the resultset. Default is a colon.

```
<paramSeparator> </paramSeparator>
```

## 2.4. paramSeparator

A character or a series of characters to use as separator when using parameters in a trigger, for example in `!show a#b#3` the `#` character is the `paramSeparator`. Default is a space.

```
<paramSeparator> </paramSeparator>
```

## 2.5. statement

Create a trigger (new command) that the bot will respond to. The plugin will perform the `SELECT` statement associated with the *trigger*. The resultset is always shown in the channel.

- **trigger** - The command which executes the SQL statement
- **query** - A `SELECT` statement

```
<statement trigger="!latest" query="SELECT subject, story FROM news WHERE id IN (SELECT max(id)
FROM news)" />
```

You may also use parameters in your queries. Put a questionmark in your query where you want the parameters to appear. This is compliant with `java.sql.PreparedStatement` usage. Furthermore you need to define each parameter and it's associated data type (valid values are string or number) and index as shown below:

```
<statement trigger="!mypastes" query="SELECT subject, l.language FROM cccp_languages l,
cccp_pastebin p WHERE p.language = l.id AND nickname=? and l.id=?">
    <column index="1" type="string" />
    <column index="2" type="number" />
</statement>
```

When using parameters, the words specified after the trigger are the parameters, for instance in `!holiday john 1 Bermuda`, `john` will be the first parameter (index 1 of type string), `1` will be the second parameter (index 2 of type number) and `Bermuda` will be the third parameter (index 3 of type string). Please note that the parameters are case sensitive.

## 3. Commands

### 3.1. Show all `SELECT` statements

**Command:** `dbq`

**Description:** Shows you a list of `SELECT` statements in the current configuration.

**Auth Level:** none

**Where to give command:** pm

**Outputs to:** DCC chat

**Example(s):**

- `/MSG Bot dbq`

### 3.2. `!latest`

**Command:** `!latest`

**Description:** This is an example command on how you can utilize this plugin.

**Auth Level:** none

**Where to give command:** channel

**Outputs to:** channel

**Example(s):**

- !latest

#### 4. Change History

v1.1	Added support for using parameters in queries. Also made more query samples
v1.0	First Version by <a href="#">far2fish</a>