

# Chart Data Error – I/O Error: Connection Reset

## Overview

A database connection pool creates and manages a pool of connections to a database. Recycling and reusing already existing connections to a database is more efficient than opening a new connection. There is one problem with connection pooling. A web application has to explicitly close the result sets, Statement's, and Connection's. Failure of a web application to close these resources can result in them never being available again for reuse those are called dead connections or a db connection pool "leak". This can eventually result in your web application connections failing if there are no more available connections. Because of these dead connections, you might see the following type of errors:

```
"An error occurred while retrieving the data for the chart XXX: Invalid state, the Connection object is closed."
```

```
"java.sql.SQLException: I/O Error: Connection reset Caused by: java.net.SocketException: Connection reset."
```

```
"[DEBUG][2008-08-26 09:39:22,395][][DataSourceMgr] getDataSource(): active connections: 0, idle connections: 3"
```

```
"[ERROR][2008-08-26 09:39:22,395][][SystemMgr] checkRepositoryTables(): ERROR CODE: 0, SQL STATE: HY010, ERROR MESSAGE: HY010 java.sql.SQLException: Invalid state, the Connection object is closed."
```

## System Configuration Error

The following error occurred while attempting to connect to the iDashboards repository database:

```
ERROR CODE: 0, SQL STATE: 08003, ERROR MESSAGE: 08003.
```

This document discusses in detail about the different reasons and resolutions for these errors. iDashboards configuration file (ivizgroup.properties) can be reviewed and revised to recover these abandoned database connections.

## Reason

Following are some of the reasons for the dead connections.

1. If the database server goes offline and the application server is not restarted.
2. If the database server is in the sleep mode.
3. The **db.maxConnections** property indicates the maximum number of connections that will be created by an iDashboards-managed connection pool. If this property is missing or blank, a default value of 20 will be used. Connections will only be created and added to the pool on an as-needed basis, so the maximum number of connections may never be reached unless the iDashboards server is extremely busy. But for some reason the **db.maxConnections** is mistyped to set to 2 instead of 20 and the iDashboards server is kept extremely busy with many users attempting to use a new connection.

---

## Recommendations

1. It is always a good practice to recycle the application server whenever you recycle the database server.
2. Make sure the database server is not in the sleep mode.
3. Increase the `db.maxConnections` property to your needs if you have large number of users using iDashboards application at a given point of time.
4. As a last resort you can always set `db.validateConnections=true` in your `ivizgroup.properties` file if the problem occurs repeatedly. The **`db.validateConnections`** property indicates whether connections should be tested when they are returned to the connection pool after use. If true, then a test query will be run on each connection when it is returned to the connection pool, and if it fails, the connection will be considered “dead” and removed from the connection pool. Since there is a small performance cost associated with testing connections, this property should be set to true only in cases where the repository database may go temporarily offline while the iDashboards application server remains online. Make sure to restart the application server after setting the following property:

```
db.validateConnections=true
```

That will validate a connection after it's used, and expunge it from the pool if the validation fails. A user might see an error, but it will work on the next try.