Command Line Interface

In addition to the DbVisualizer GUI tool, there is also a pure command line interface for running scripts. We recommend that you use this interface for tasks that you schedule via the operating system's scheduling tool, or when you need to include database tasks in a command script for a larger job. It is also the right tool for execution of large scripts, such as a script generated by the DbVisualizer Export Schema feature.

On Windows and Linux/Unix, you find this command as a BAT file (*dbviscmd.bat*) or a shell script (*dbviscmd.sh*) in the DbVisualizer installation directory. For macOS, the shell script is located in */Application/DbVisualizer-<Version>.app/Contents/Resources/app*.

Command Line Options

The command line interface supports the following options:

```
 Usage: dbviscmd (-connection <name> |
                -url -drivername | -url -driverclass -driverpath))
                [-userid <userid>] [-password <password>]
                -sql <statements> | -sqlfile <filename> [-encoding <encoding>]
                [-catalog <catalog>] [-schema <schema>]
                [-maxrows <max>] [-maxchars <max>]
                [-stoponerror] [-stoponwarning]
                [-stripcomments true | false]
                [-output all | none | log | result] [-outputfile <filename>]
                [-listconnections]
                [-debug [-debugfile <filename>]
                [-prefsdir <directory>]
                [-masterpw <password>] [-help] [-version]

Options:
 -connection <name>     Database connection name (created with the GUI)
 -url <URL>             Database URL
 -drivername <name>     Database driver name (created with the GUI)
 -driverclass <name>    Full name of the JDBC Driver class name
 -driverpath <p1:p2..>  Paths to the jar files constituting the JDBC driver. Each path separated by a ":"
 -userid <userid>       Userid to connect as
 -password <password>   Password for userid
 -sql <statements>      One or more delimited SQL statements
 -sqlfile <filename>    SQL script file to execute
 -encoding <encoding>   Encoding for the SQL script file
 -catalog <catalog>     Catalog to use for unqualified identifiers
 -schema <schema>       Schema to use for unqualified identifiers
 -maxrows <max>         Maximum number of rows to display for a result set
 -maxchars <max>        Maximum number of characters to display for a column
 -stoponerror           Stop execution when getting an error
 -stoponsqlwarning      Stop execution when getting an SQL warning
 -stoponnorows          Stop execution when empty result set or no affected rows
 -stripcomments         Strip comments before sending to database. Default is
                        the setting made in the GUI
 -output                "all" (default), output both log msgs and result sets
                        "none", suppress both log messages and result sets
                        "log", output only log messages
                        "result", output only result sets
 -outputfile <filename> Script execution output file. Default is stdout
 -listconnections       Lists all database connections
 -debug                 Write debug messages
 -debugfile <filename>  File for debug messages. Default is stderr
 -errordir <directory>  Use an alternate location for error logs
 -prefsdir <directory>  Use an alternate user preferences directory
 -masterpw <password>   Master Password for encrypted database passwords
 -help                  Display this help
 -version               Show version info
```

There are two options to to specify which database to connect to:

- Using a connection already defined by the DbVisualizer tool. This is done by using the `-connection` parameter. If you have forgot the connection name, use the `-listconnections` option to get a list of all existing names.
- Specifying the connection properties by using the parameter `-url`. The `-url` parameter specifies the JDBC URL for the database to connect to. For information about the JDBC url see Setting Up a Connection Manually. There are also some examples below showing how to specify connection properties using the `-url` parameter

Examples

Executing single statements

You can use the command line interface to execute a single SQL statement:

```
> dbviscmd.bat -connection "Oracle" -sql "select * from hr.countries"
14:34:48  START Executing Command Line, Database Connection: Oracle Database Type: ORACLE Catalog: null Schema:
SYSTEM
14:34:48  INFO  Physical database connection acquired for: Oracle
COUNTRY_ID  COUNTRY_NAME              REGION_ID
----------  ------------------------  ---------
AR          Argentina                 2
AU          Australia                 3
BE          Belgium                   1
BR          Brazil                    2
CA          Canada                    2
CH          Switzerland               1
CN          China                     3
DE          Germany                   1
DK          Denmark                   1
EG          Egypt                     4
FR          France                    1
HK          HongKong                  3
IL          Israel                    4
IN          India                     3
IT          Italy                     1
JP          Japan                     3
KW          Kuwait                    4
MX          Mexico                    2
NG          Nigeria                   4
NL          Netherlands               1
SG          Singapore                 3
UK          United Kingdom            1
US          United States of America  2
ZM          Zambia                    4
ZW          Zimbabwe                  4
14:34:48  SUCCESS  [SELECT - 25 rows, 0.007 secs]  Result set fetched
select * from hr.countries;
14:34:48  END Execution 1 statement(s) executed, 25 row(s) affected, exec/fetch time: 0.007/0.005 secs   [1
successful, 0 errors]
```

If you like to execute just a few statements, you can pass in a list of statements:

```
> dbviscmd.bat -connection "Oracle" -sql "select * from hr.countries; select * from hr.regions"
14:42:21  START Executing Command Line, Database Connection: Oracle Database Type: ORACLE Catalog: null Schema:
SYSTEM
14:42:21  INFO  Physical database connection acquired for: Oracle
COUNTRY_ID  COUNTRY_NAME              REGION_ID
----------  ------------------------  ---------
AR          Argentina                 2
AU          Australia                 3
BE          Belgium                   1
BR          Brazil                    2
CA          Canada                    2
CH          Switzerland               1
CN          China                     3
DE          Germany                   1
DK          Denmark                   1
EG          Egypt                     4
FR          France                    1
HK          HongKong                  3
IL          Israel                    4
IN          India                     3
IT          Italy                     1
JP          Japan                     3
KW          Kuwait                    4
MX          Mexico                    2
NG          Nigeria                   4
NL          Netherlands               1
SG          Singapore                 3
UK          United Kingdom            1
US          United States of America  2
ZM          Zambia                    4
ZW          Zimbabwe                  4
14:42:21  SUCCESS  [SELECT - 25 rows, 0.004 secs]  Result set fetched
select * from hr.countries;
REGION_ID  REGION_NAME
---------  ----------------------
5          Australia
6          South America
1          Europe
2          Americas
3          Asia
4          Middle East and Africa
14:42:21  SUCCESS  [SELECT - 6 rows, 0.003 secs]  Result set fetched
select * from hr.regions;
14:42:21  END Execution 2 statement(s) executed, 31 row(s) affected, exec/fetch time: 0.007/0.002 secs   [2
successful, 0 errors]
```

Executing scripts

If you frequently want to execute a number of statements, it's best to put them into a script file. Here's how to execute a script that contains the two statements from the example above:

```
> dbviscmd.bat -connection "Oracle" -sqlfile "myscript.sql"

14:45:11  START Executing Command Line, Database Connection: Oracle Database Type: ORACLE Catalog: null Schema:
SYSTEM
14:45:11  INFO  Physical database connection acquired for: Oracle
COUNTRY_ID  COUNTRY_NAME              REGION_ID
----------  ------------------------  ---------
AR          Argentina                 2
AU          Australia                 3
BE          Belgium                   1
BR          Brazil                    2
CA          Canada                    2
CH          Switzerland               1
CN          China                     3
DE          Germany                   1
DK          Denmark                   1
EG          Egypt                     4
FR          France                    1
HK          HongKong                  3
IL          Israel                    4
IN          India                     3
IT          Italy                     1
JP          Japan                     3
KW          Kuwait                    4
MX          Mexico                    2
NG          Nigeria                   4
NL          Netherlands               1
SG          Singapore                 3
UK          United Kingdom            1
US          United States of America  2
ZM          Zambia                    4
ZW          Zimbabwe                  4
14:45:11  SUCCESS  [SELECT - 25 rows, 0.004 secs]  Result set fetched
select * from hr.countries;
REGION_ID  REGION_NAME
---------  ----------------------
5          Australia
6          South America
1          Europe
2          Americas
3          Asia
4          Middle East and Africa
14:45:11  SUCCESS  [SELECT - 6 rows, 0.003 secs]  Result set fetched
select * from hr.regions;
14:45:11  END Execution 2 statement(s) executed, 31 row(s) affected, exec/fetch time: 0.007/0.002 secs   [2
successful, 0 errors]
```

Controlling the output

You can use options to control how much output to generate. If you only want to see the results, use the `-output` option with the result keyword:

```
> dbviscmd.bat -connection "Oracle" -sqlfile "myscript.sql" -output result
COUNTRY_ID  COUNTRY_NAME              REGION_ID
----------  ------------------------  ---------
AR          Argentina                 2
AU          Australia                 3
BE          Belgium                   1
BR          Brazil                    2
CA          Canada                    2
CH          Switzerland               1
CN          China                     3
DE          Germany                   1
DK          Denmark                   1
EG          Egypt                     4
FR          France                    1
HK          HongKong                  3
IL          Israel                    4
IN          India                     3
IT          Italy                     1
JP          Japan                     3
KW          Kuwait                    4
MX          Mexico                    2
NG          Nigeria                   4
NL          Netherlands               1
SG          Singapore                 3
UK          United Kingdom            1
US          United States of America  2
ZM          Zambia                    4
ZW          Zimbabwe                  4
REGION_ID  REGION_NAME
---------  ----------------------
1          Europe
2          Americas
3          Asia
4          Middle East and Africa
```

For other scripts, for instance a script containing INSERT statements, you may only want to see the log messages:

```
> dbviscmd.bat -connection "Oracle" -sqlfile "myscript.sql" -output log
14:25:29  START Executing Command Line, Database Connection: Oracle Database Type: ORACLE Catalog: null Schema:
SYSTEM
14:25:30  INFO  Physical database connection acquired for: Oracle
14:25:30  SUCCESS  [SELECT - 25 rows, 0.012 secs]  Result set fetched
select * from hr.countries;
14:25:30  SUCCESS  [SELECT - 4 rows, 0.009 secs]  Result set fetched
select * from hr.regions;
14:25:30  END Execution 2 statement(s) executed, 29 row(s) affected, exec/fetch time: 0.021/0.004 secs   [2
successful, 0 errors]
```

Combining OS scripts, the command line interface and DbVisualizer variables

For more complex tasks, you can call the command line interface from a shell script, for instance a Bourne shell script on Unix or a BAT file on Windows. You can also use DbVisualizer variables to pass information between the shell script and the SQL script. In this example, we have a simple SQL script (*cmdtest.sql*) that contains a SELECT statement with a variable in place for the table name:

cmdtest.sql

```
select * from ${table}$
```

A text file (*tables.txt*) contains the table names we want to execute the SQL script with:

tables.txt

```
hr.countries
hr.regions
```

In a command shell (Bourne or Bash), we can then execute the script using the table names from the text file:

```
for name in `cat tables.txt`;
  do ./dbviscmd.sh -connection "oracle" -sql "@run cmdtest.sql \${table||$name||||nobind}\$; ";
done

15:01:19  START Executing Command Line, Database Connection: oracle Database Type: ORACLE Catalog: null Schema:
SYSTEM
15:01:20  INFO  Physical database connection acquired for: oracle
15:01:20  RUNNING  [@run ...ntries||||nobind}$ - - secs]
@run cmdtest.sql ${table||hr.countries||||nobind}$;
COUNTRY_ID  COUNTRY_NAME              REGION_ID
----------  ------------------------  ---------
AR          Argentina                 2
AU          Australia                 3
BE          Belgium                   1
BR          Brazil                    2
CA          Canada                    2
CH          Switzerland               1
CN          China                     3
DE          Germany                   1
DK          Denmark                   1
EG          Egypt                     4
FR          France                    1
HK          HongKong                  3
IL          Israel                    4
IN          India                     3
IT          Italy                     1
JP          Japan                     3
KW          Kuwait                    4
MX          Mexico                    2
NG          Nigeria                   4
NL          Netherlands               1
SG          Singapore                 3
UK          United Kingdom            1
US          United States of America  2
ZM          Zambia                    4
ZW          Zimbabwe                  4
15:01:20  SUCCESS  [SELECT - 25 rows, 0.016 secs]  Result set fetched
select * from hr.countries;
15:01:20  SUCCESS  [@run ...ntries||||nobind}$ - 0.016 secs]  Script processed
@run cmdtest.sql ${table||hr.countries||||nobind}$;
15:01:20  END Execution 1 statement(s) executed, 25 row(s) affected, exec/fetch time: 0.016/0.012 secs   [1
successful, 0 errors]
java -cp /Users/ulf/work/github/dbvis/trunk/pureit/apps/dbvis/classes:/Users/ulf/work/github/dbvis/trunk/pureit
/apps/dbvis/resources:/Users/ulf/work/github/dbvis/trunk/pureit/apps/dbvis/external/* -Xmx512M -Djava.awt.
headless=true -Ddbvis.home=/Users/ulf/work/github/dbvis/trunk/pureit/apps/dbvis com.onseven.dbvis.
DbVisualizerCmd -masterpw stairway -connection oracle -sql @run cmdtest.sql ${table||hr.regions||||nobind}$;
15:01:23  START Executing Command Line, Database Connection: oracle Database Type: ORACLE Catalog: null Schema:
SYSTEM
15:01:23  INFO  Physical database connection acquired for: oracle
15:01:23  RUNNING  [@run ...egions||||nobind}$ - - secs]
@run cmdtest.sql ${table||hr.regions||||nobind}$;
REGION_ID  REGION_NAME
---------  ----------------------
5          Australia
6          South America
1          Europe
2          Americas
3          Asia
4          Middle East and Africa
15:01:23  SUCCESS  [SELECT - 6 rows, 0.001 secs]  Result set fetched
select * from hr.regions;
15:01:23  SUCCESS  [@run ...egions||||nobind}$ - 0.001 secs]  Script processed
@run cmdtest.sql ${table||hr.regions||||nobind}$;
15:01:23  END Execution 1 statement(s) executed, 6 row(s) affected, exec/fetch time: 0.001/0.000 secs   [1
successful, 0 errors]
```

The command line interface is called with the `-sql` option, specifying the client-side command `@run`. A DbVisualizer variable is passed to the `@run` command with the value taken from the shell variable. This DbVisualizer variable value is then available to the script executed by the `@run` command.

Note that you may need to escape certain characters that the shell would otherwise interpret, like the dollar signs that are part of the DbVisualizer variable delimiters.

Setting up the connection properties on the command line

As an alternative to using a connection already set-up through the DbVisualizer tool you may use the `-url` parameter. In combination with the parameters `-drivername`, `-driverclass`, and `-driverpath` these parameters enables you connect without prior specification using the DbVisualizer tool.

Following are some examples.

Executing SQL towards a MySQL instance running on `localhost` port `3306`. The parameter `"-drivername MYSQL"` specifies that we are using a JDBC driver specified in the DbVisualizer tool named `MYSQL`. For listing of the existing drivers use the **Tools->Driver Manager** in the DbVisualizer tool. Read more in Installing a JDBC Driver.

Using -url and -drivername parameters

```
./dbviscmd.sh -url jdbc:mysql://localhost:3306/
   -drivername MYSQL
   -sql "select * from sakila.actor"
   -userid root
```

An alternative to use the `-drivername` you may use the parameters `-driverclass` and `-driverpath` to specify the JDBC driver.

Using -driverclass and -driverpath parameters

```
./dbviscmd.sh -url jdbc:oracle:thin:@localhost:1521/ORCL
   -driverclass oracle.jdbc.OracleDriver
   -driverpath "ojdbc6.jar:orai18n.jar:xdb.jar:xmlparserv2.jar"
   -sql "select * from HR.COUNTRIES"
   -userid system
   -password oracle
```

The above example connects to an Oracle instance on `localhost` and port `1521`. Note that the separator character **':'** between the different jar files is platform dependant. On Windows-based desktop platforms, the value of this field is the semicolon **";"**.