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 Mac OS X, 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> [-userid <userid>] [-password <password>]
                -sql <statements> | -sqlfile <filename> [-encoding <encoding>]
                [-catalog <catalog>] [-schema <schema>]
                [-maxrows <max>] [-maxchars <max>]
                [-stoponerror] [-stoponwarning]
                [-output all | none | log | result] [-outputfile <filename>]
                [-listconnections]
                [-debug [-debugfile <filename>]
                [-prefsdir <directory>] [-help] [-version]
                               [-masterpw <password>]

Options:
 -connection <name>     Database connection name (created with the GUI)
 -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
 -stoponwarning         Stop execution when getting a warning
 -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
 -prefsdir <directory>  Use an alternate user preferences directory
 -masterpw <password>   Master Password for encrypted database passwords
 -help                  Display this help
 -version               Show version info
```

Before you can use the command line tool, you need to create at least one database connection using the GUI tool. You need to specify the connection to use with the `-connection` option when you run `dbviscmd`. If you have forgot the connection name, use the `-listconnections` option to get a list of all connection names.

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"
select * from hr.countries;
INFO: 14:20:31  [SELECT - 25 row(s), 0.247 secs] Result set fetched
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
SUMMARY: ... 1 statement(s) executed, 25 row(s) affected, exec/fetch time: 0.247/0.002 sec
 [1 successful, 0 warnings, 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"
select * from hr.countries;
INFO: 14:23:39  [SELECT - 25 row(s), 0.012 secs] Result set fetched
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
select * from hr.regions;
INFO: 14:23:39  [SELECT - 4 row(s), 0.130 secs] Result set fetched
REGION_ID  REGION_NAME
---------  ----------------------
1          Europe
2          Americas
3          Asia
4          Middle East and Africa
SUMMARY: ... 2 statement(s) executed, 29 row(s) affected, exec/fetch time: 0.142/0.003 sec
 [2 successful, 0 warnings, 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"
select * from hr.countries;
INFO: 16:38:06  [SELECT - 25 row(s), 0.021 secs] Result set fetched
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
select * from hr.regions;
INFO: 16:38:06  [SELECT - 4 row(s), 0.005 secs] Result set fetched
REGION_ID  REGION_NAME
---------  ----------------------
1          Europe
2          Americas
3          Asia
4          Middle East and Africa
SUMMARY: ... 2 statement(s) executed, 29 row(s) affected, exec/fetch time: 0.026/0.001 sec
 [2 successful, 0 warnings, 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
select * from hr.countries;
INFO: 16:52:56  [SELECT - 25 row(s), 0.013 secs] Result set fetched
select * from hr.regions;
INFO: 16:52:56  [SELECT - 4 row(s), 0.002 secs] Result set fetched
SUMMARY: ... 2 statement(s) executed, 29 row(s) affected, exec/fetch time: 0.015/0.002 sec
 [2 successful, 0 warnings, 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

@run /Users/hans/tmp/cmdtest.sql ${table||hr.countries||||nobind}$;
INFO: 17:08:16  [@RUN - 0 row(s), 0.000 secs] Command processed
select * from hr.countries;
INFO: 17:08:16  [SELECT - 25 row(s), 0.012 secs] Result set fetched
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
SUMMARY: ... 2 statement(s) executed, 25 row(s) affected, exec/fetch time: 0.012/0.002 sec
 [2 successful, 0 warnings, 0 errors]
@run /Users/hans/tmp/cmdtest.sql ${table||hr.regions||||nobind}$;
INFO: 17:08:18  [@RUN - 0 row(s), 0.000 secs] Command processed
select * from hr.regions;
INFO: 17:08:18  [SELECT - 4 row(s), 0.013 secs] Result set fetched
REGION_ID  REGION_NAME
---------  ----------------------
1          Europe
2          Americas
3          Asia
4          Middle East and Africa
SUMMARY: ... 2 statement(s) executed, 4 row(s) affected, exec/fetch time: 0.013/0.000 sec
 [2 successful, 0 warnings, 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.