

Q-Checker database connection (jdbc-booking)

The following files has to be adapt:

1.) **qcheckerV5.vbs / qcheckerV5 / qcheckerV4**

In the Q-Checker script the paths of the java-files have to be set. The name of the JAVA_JDBC-file is different in reference to the used database (DB2, Oracle, MySQL, postgresql, ...).

```
QCHECKER_DB_METHOD = "JDBC"  
...  
QCHECKER_JAVA_CLASSES = "C:\IFOR\JRE1.3.0\JRE\LIB\rt.jar"  
QCHECKER_JAVA_BOOK = QCHECKER_LOAD_JAVA & "\Q-CheckerDB.jar"  
QCHECKER_JAVA_JDBC = QCHECKER_LOAD_JAVA & "\postgresql.jar"  
QCHECKER_JAVA_CALL = "C:\IFOR\JRE1.3.0\JRE\BIN\javaw.exe"
```

2.) **CATIA-Q-Checker-Environment.txt / QCHECKER.dcls**

Q-Checker V5

In the CATIA-Q-Checker-Environment.txt the variable QCLICDB has to be set to YES in case to use the database-license (TC-qcheckerV5-DB).

```
QCLICDB=YES
```

Q-Checker V4

In the QCHECKER.dcls the variable QCLICDB has to be set to YES in case to use the database-license (TC-qchecker-all-DB).

```
alias QCLICDB = catia.QCHECKER_LICENSE_DATABASE='YES';
```

3.) **QCHECKER.par**

To create the xml-report the Q-Checker-configuration-file has to be adapted.

```
qchecker.DB_CONNECT_BATCH          YES  
...  
qchecker.DB_CONNECT_INTERACTIVE    YES
```

4.) **QCHECKER.db**

Adapt the database-configuration-file for the used database.

```
# Define database type  
qchecker.DB_TYPE          POSTGRESQL  
  
# Define user with write access to database  
qchecker.DB_USER          qchecker  
  
# Define password of user with write access to database  
qchecker.DB_PASSWORD      qchecker  
  
# Define address of database server  
qchecker.DB_SERVER        pcprog3  
  
# Define database name  
qchecker.DB_NAME          qc_v5  
  
# Define port for remote database access  
qchecker.DB_PORT          5432
```

Possible verifications:

- License:

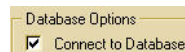
Q-Checker can be started interactively?

Info: The license is checked during the Q-Checker start. If the license is not available Q-Checker will not start and a corresponding error-message will appear. Reverse: If Q-Checker can be launched, the license is available.

Do the Database Options can be activated in the panel Batch-Settings?

Q-Checker V5

Info: The use of the database-license is controlled by a variable in the CATIA-Q-Checker-Environment.txt. If the variable „QCLICDB=YES“ is set, the Database Options can be activated in the panel Batch-Settings. The database-license is available.



If the variable „QCLICDB=NO“ is set, the Database Options can not be activated in the panel Batch-Settings. The database-license is not available.



Q-Checker V4

Analog Q-CheckerV5. The use of the database-license is controlled by a variable in the QCHECKER.dcls.

- QCHECKER-par

Do Q-Checker write the xml-reports?


Info: If the QCHECKER.par is well configured - and the database-license is available (see above) - in addition to the qcreport-file, the xml-report-file is written after each interactive and / or batch-run. The xml-report contains all information that are necessary for the database booking. That are, beside the data itself, the information from the QCHECKER.db.

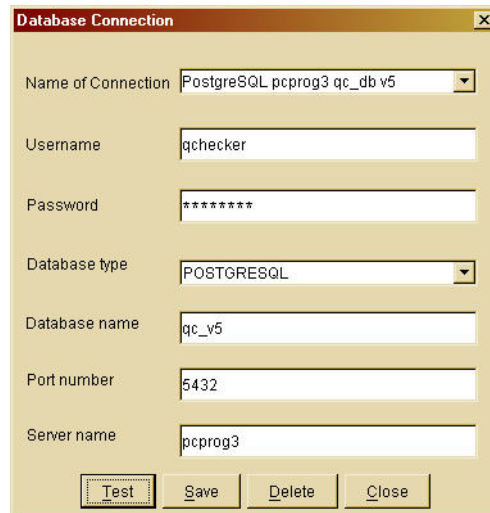
The files are saved into the report-directory which is defined in the environment by the variable „QCREPORT=path“, resp. in the QCHECKER.dcls by the variable” alias QCREPORT = catia.QCHECKER_REPORT='path'”

Note: In opposition to the *.xml the *.qcanalysis.xml is not used for the database booking but for the visualization in the Q-Checker Analysis Viewer.

- QCHECKER.db

Are the database-information correct?

Info: In Q-Monitor information about the database-connection can be displayed with the button „connect“  .



The information in the DB_INFO-block of the xml-report must concur with that in Database Connection of Q-Monitor.

```
<DB_INFO>
  <DB_TYPE> 'POSTGRESQL' </DB_TYPE>
  <DB_USER> 'qchecker' </DB_USER>
  <DB_PASSWORD> 'qchecker' </DB_PASSWORD>
  <DB_SERVER> 'pcprog3' </DB_SERVER>
  <DB_NAME> 'qc_v5' </DB_NAME>
  <DB_PORT> '5432' </DB_PORT>
</DB_INFO>
```

If the DB_INFO-Block is empty two reasons are possible:

- The QCHECKER.db does not exist.
- During the saving of the adapted file with a text-editor a .txt as been added. Some editors do this automatically. The file QCHECKER.db.txt is not identified by Q-Checker.

In both cases a new (empty) QCHECKER.db is created by Q-Checker in the db-folder.

- **qcheckerV5.vbs**

Do the correct JAVA_JDBC-file is used?

Info: The JAVA_JDBC-file depends on the used database. Therefore the Q-Monitor is suitable to check if the used file is correct. Click on the „Test“ button in the panel „Database Connection“ (see above). If the message „Connection OK“ is displayed the file, entered in the qmonitor.bat, is correct.



qmonitor.bat

```
start C:\Programme\JavaSoft\JRE\1.3.1\bin\java.exe -classpath
.\qmonitor.2.5.8.jar;postgresql.jar qmon.QMonitor .\qmonitor.ini
```

Save a copy of this file in the load-directory of Q-Checker and adapt the variable QCHECKER_JAVA_JDBC in the qcheckerV5.vbs (resp. qcheckerV5, qcheckerV4).

```
QCHECKER_JAVA_JDBC = QCHECKER_LOAD_JAVA & "postgresql.jar "
```