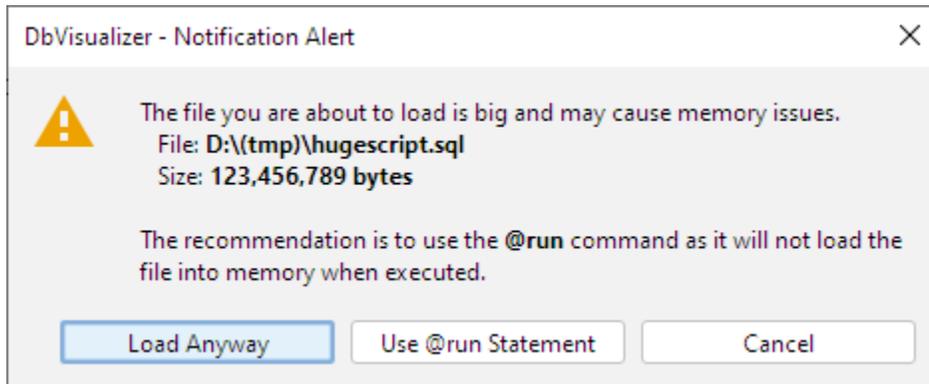# Executing an External Script

If you have a very large script to execute, you may not have enough memory available to be allocated for DbVisualizer to load it into an SQL Commander editor.

To save memory, you can use the **@run** command. If you try to load a very large file, DbVisualizer suggests using the @run command automatically:

The **@run** command executes a script file by only loading one statement at a time, minimizing the memory requirements. A related command is the **@cd** command for changing the current directory.

- **@run <file> [<variables>]**
  Request to execute the file specified as parameter, optionally with a list of variables
- **@cd <directory>**
  Change the working directory for the following **@run** command

Here's an example of a script using these commands:

```
@run createDB.sql;        -- Execute the content in the
                          -- createDB.sql file without loading into the SQL editor.
                          -- The location of this file is the same as the working
                          -- directory for DbVisualizer (when not using an absolut path).
@cd /home/mupp;           -- Request to change directory to /home/mupp
@cd myscripts;            -- Request to change directory relative to current, i.e. to /home/mupp/myscripts
@run loadBackup.sql;      -- Execute the content in the loadBackup.sql
                          -- file relative to current directory. This file will now be read from the
                          -- /home/mupp/myscripts directory.
```

You can also include DbVisualizer variables as parameters to the **@run** command, with values to be used for the corresponding variables in the script:

```
@run monthlyReport ${month||2021-01-14||Date||noshow}$ ${dept||HR||String||noshow}$
```

Even though the **@run** command reads one statement at a time from the file, there are other parts of the execution process that require the whole file to be read before the statements can be executed: parsing the script for variables, parameter markers, and restricted commands, as well as counting all statements in order to provide progress information. When you run a script that is large enough (more than 10 MB) for these things to potentially cause memory problems and slow down the processing, DbVisualizer gives you a chance to turn off this preprocessing and progress reporting so that the statements instead can be executed directly as the are read from the file, one at a time.
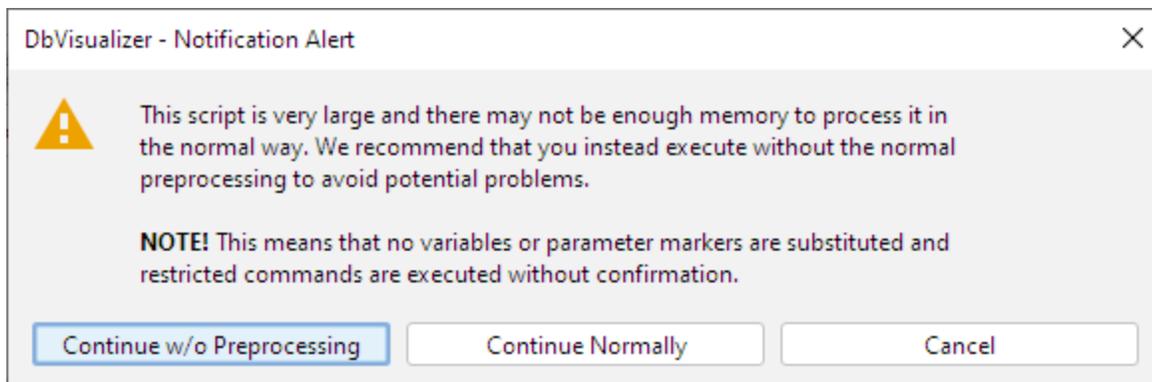
To ensure that you don't have any problems running scripts this large we strongly recommend that you click **Continue w/o Preprocessing**, thereby disabling all variable, parameter and restricted commands processing. Only click **Continue Normally** if you know for sure that you have enough memory available and have adjusted your installation so that DbVisualizer can use it. With the preprocessing disabled, you should be able to execute scripts of any size (we have tested with scripts as large as 4 GB).

Another alternative for execution of large scripts is to use the DbVisualizer command line interface instead of the GUI application. This option is the most efficient and fastest.

Running without preprocessing is always more efficient, so if your script does not use any variables or parameter markers and you do not use the Permissions feature, you can disable preprocessing even for scripts smaller than 10 MB by unchecking the **SQL Commander Options->Preprocess Script** checkbox in the SQL Commander menu.