


Exporting Query Results

Only in DbVisualizer Pro

 This feature is only available in the DbVisualizer Pro edition.

Instead of viewing query results in Result Set grids, you can export the result of one or more queries to a file. For very large results, this may be the preferred choice due to memory constraints.

To export a query result, create a script with

1. an **@export on** command,
2. an **@export set** command,
3. one or more queries,
4. an **@export off** command.

Here is a basic example:

```
@export on;
@export set filename="c:\Backups\Orders.csv";
select * from Orders;
@export off;
```

The **@export set** command takes a parameter name followed by an equal sign and a value. You can use the following parameters, where only filename is required and all names are case-insensitive:

Parameter	Default	Valid Values
AppendFile	false	true, false, clear (i.e. start with a new file for the first result and then append to it)
BinaryFileDir		Path for data files when BinaryFormat is set to File , see Exporting a Table for details. Note! Variables for column names must include the scope option when entered manually into a script, e.g. <code>/Users/hans/exp/\${dbvis-date}/\${COUNTRIES} scope=post}\$</code>
BinaryFormat	Don't Export	Don't Export, Size, Value, Hex, Base64, File
BooleanFalseFormat	false	false, no, 0, off
BooleanTrueFormat	true	true, yes, 1, on
CLOBFileDir		Path for data files when CLOBFormat is set to File , see Exporting a Table for details. Note! Variables for column names must include the scope option when entered manually into a script, e.g. <code>/Users/hans/exp/\${dbvis-date}/\${COUNTRIES} scope=post}\$</code>
CLOBFormat	Don't Export	Don't Export, Size, Value, File
CsvColumnDelimiter	\t (TAB)	
CsvIncludeColumnHeader	true	true, false
CsvIncludeColumnHeaderPerResult	false	
CsvColumnHeaderIsColumnAlias	true	true, false
CsvIncludeSQLCommand	Don't Include	Don't Include, Top, Bottom
CsvSplitFileSize	-1	Split the result over multiple files if it is larger than the specified size, or -1 to never split. The size must be specified as <code>size [g G m M k K]</code>
CsvRemoveNewlines	false	true, false
CsvRowCommentIdentifier		

CsvRowDelimiter	\n	\n (UNIX/Linux/macOS), \r\n (Windows)						
DateFormat	yyyy-MM-dd	See valid formats in Changing the Data Display Format document						
DecimalNumberFormat	Unformatted	See valid formats in Changing the Data Display Format document						
Destination	File	File						
Encoding	UTF-8	Check supported encodings for all encodings. (Use the encoding in the Canonical Name for java.nio API column).						
ExcelColumnHeaderIsColumnAlias	true	true, false						
ExcelFileFormat	Binary	Two formats are supported <ul style="list-style-type: none"> Binary Excel (xls) supporting export of max 65 535 rows. Specified as "Binary Excel (xls)", "Binary" or "xls" OOXML, Excel 2007 (xlsx). Specified as "OOXML, Excel 2007 (xlsx)", "OOXML" or "xlsx" 						
ExcelIncludeColumnHeader	true	true, false						
ExcelIncludeSQLCommand	false	true, false						
ExcelIntroText		Any description						
ExcelSheetName		Used when exporting to excel. Sets the name of exported excel sheet.						
ExcelTextOnly	false	true, false. Convert numeric values to text in the Excel file if true.						
ExcelTextDateTime	true	true, false. Convert date, time and timestamp data to text in the Excel file if true.						
ExcelTitle		Any title						
Filename	REQUIRED							
Format	Based on file extension, or CSV if none	CSV, HTML, XML, SQL, XLS, JSON. If Format is not specified, the file extension is used to determine the format. If there is no recognized file extension, CSV is used as the default.						
HtmlColumnHeaderIsColumnAlias	true	true, false						
HtmlConvertChars	true	true, false. Set to false if you have HTML code in the exported data, so that e.g. < and > characters are not converted to < and >						
HtmlFooter	[Generated by: DbVisualizer <i>version</i>]	Any text to use in the document footer. Can be set to blank to remove the footer.						
HtmlIncludeSQLCommand	false	true, false						
HtmlIntroText		Any description						
HtmlPerTableHeader	E.g. <table border="1" data-bbox="277 1705 570 1892"> <tr> <td>Date:</td> <td>2017-05-31 16:48:23</td> </tr> <tr> <td>Columns:</td> <td>4</td> </tr> <tr> <td>Table:</td> <td>COUNTRIES</td> </tr> </table>	Date:	2017-05-31 16:48:23	Columns:	4	Table:	COUNTRIES	HTML code that describes the table. To fit into the rest of the HTML code, it must start with <tr> and end with </tr>. The pre-defined DbVisualizer variables can be used, e.g. \${dbvis-object}\$ to include the table name.
Date:	2017-05-31 16:48:23							
Columns:	4							
Table:	COUNTRIES							
HtmlTitle		Any title						

JSONColumnHeaderIsColumnAlias	true	true, false
JSONStyle	Array	Array, Rows
NumberFormat	Unformatted	See valid formats in Changing the Data Display Format document
QuoteDuplicateEmbedded	true	true, false (quote char is the same as QuoteTextData)
QuoteTextData	None	None, Single, Double
Settings		The path to an XML file containing all settings
ShowNullAs	(null)	
SqlAfterExportStmts		Any statements to include in the script after the SQL statements for the exported objects, e.g. <code>set foreign_key_checks = 1;</code>
SqlBeforeExportStmts		Any statements to include in the script before the SQL statements for the exported objects, e.g. <code>set foreign_key_checks = 0;</code>
SqlBeginIdentifier		Character to use to begin a quoted identifier. Note! To specify a double-quote, you must duplicate it since double-quote is also used to enclose the parameter value.
SqlBlockBeginDelim		String to use to begin an SQL block when exporting complex DDL statements using the <code>@ddl</code> command.
SqlBlockEndDelim		String to use to end an SQL block
SqlDelimitedIdentifiers	true	true, false
SqlEndIdentifier		Character to use to end a quoted identifier. Note! To specify a double-quote, you must duplicate it since double-quote is also used to enclose the parameter value.
SqlGroupBy	Object	Object, Statement. Set to Object to generate DROP, CREATE, INSERT, and ALTER statements (where applicable) for each exported object in turn. Set to Statement to group all statements of the same type together, e.g. first DROP statements for all exported objects, then CREATE statements for all exported objects, etc.
SqlIncludeAutoGeneratedValues	true	Set to false to exclude columns declared as AUTO_INCREMENT or IDENTITY in the INSERT statements.
SqlIncludeCreateDDL	false	true, false
SqlIncludeSQLCommand	Don't Include	Don't Include, Top, Bottom
SqlQualifier		Qualifier to use when qualifying table names. If not set, DbVisualizer tries to determine the schema and use it as the qualifier.
SqlQualifyColumnName	false	true, false
SqlQualifyObjectName	true	true, false
SqlRowCommentIdentifier	--	
SqlSeparator	;	Statement separator character.
SqlSplitFileSize	-1	Split the result over multiple files if it is larger than the specified size, or -1 to never split. The size must be specified as <code>size [g G m M k K]</code>
TableName		Can be set if DbVisualizer cannot determine the value for the <code>\$(dbvis-object)\$</code> variable
TimeFormat	HH:mm:ss	See valid formats in Changing the Data Display Format document
TimeStampFormat	yyyy-MM-dd HH:mm:ss.SSSSSS	See valid formats in Changing the Data Display Format document

XmlColumnHeaderIsColumnAlias	true	true, false
XmlIncludeSQLCommand	false	true, false
XmlIntroText		Any description
XmlStyle	DbVisualizer	DbVisualizer, XmlDataSet, FlatXmlDataSet

Here are a few examples using some of these settings.

Automatic table name to file mapping

This example shows how to make the filename the same as the table name in the select statement. The example also shows several select statements. Each will be exported in the SQL format. Since the filename is defined to be automatically set, this means that there will be one file per result set and each file is named by the name of its table.

```
@export on;
@export set filename="c:\Backups\${dbvis-object}%" format="sql";
select * from Orders;
select * from Products;
select * from Transactions;
@export off;
```

There must be only one table name in a select statement in order to automatically set the filename with the **`\${dbvis-object} %`** variable, i.e if the select joins from several tables or pseudo tables are used, you must explicitly name the file.

The **`\${dbvis-object} %`** variable is not substituted with a table name if using the AppendFile="true/clear" parameter.

Multiple results to a single file

This example shows how all result sets can be exported to a single file. The **AppendFile** parameter supports the following values.

- **true**
The following result sets will all be exported to a single file
- **false**
Turn off the append processing
- **clear**
Same as the true value but this will in addition clear the file before the first result set is exported

```
@export on;
@export set filename="c:\Backups\alltables.sql" appendfile="clear" format="sql";
select * from Orders;
select * from Products;
select * from Transactions;
@export off;
```

Using predefined settings

If you save settings when [exporting a table](#) or a [schema](#), you can use the **Settings** parameter to reference the settings file.

```
@export on;
@export set settings="c:\tmp\htmlsettings.xml" filename="c:\Backups\${dbvis-object} %";
select * from Orders;
select * from Products;
select * from Transactions;
@export off;
```

Limit the number of exported rows

You can use the `@set maxrows` command in combination with the `@export` command to override the [Max Rows](#) field value in the SQL Commander tab toolbar.

```
@set maxrows 10;
@export on;
@export set filename="c:\Backups\alltables.sql" format="sql";
select * from Orders;
select * from Products;
select * from Transactions;
@export off;
```

If Max Rows is set to a positive number, you can use the `@set maxrows` command to set it to `-1` to export all rows.

Don't Export