

SQLScriptWriter Help Contents

[About SQLScriptWriter](#)

[Getting Started](#)

[Registering SQLScriptWriter](#)

[Working With Databases](#)

[Executing Your Script](#)

[Text Editing](#)

[Using Metadata](#)

[Printing](#)

[Customising SQLScriptWriter](#)

Copyright ©1996 [Neil Butterworth](#) .

All trademarks and registered trademarks acknowledged.

About SQLScriptWriter

SQLScriptWriter is a combined editor and execution environment, running under 32-bit Windows, for developing SQL DDL and DML scripts. For portability, it accesses your database via the Borland Database Engine architecture.

SQLScriptWriter offers many productivity features, some of which include:

- A combined edit and execute environment which means no switching between tools as you develop your scripts, and which allows you to develop your SQL scripts in an iterative, RAD-like way.
- The ability to view database data and metadata while editing scripts, and to insert database object names into your script using either a simple point-and-click interface, or name completion within the editing environment.
- Macros, which allow you to perform frequently performed scripting tasks such as view creation with a single mouse click or via an editor abbreviation.
- SQL keyword completion, putting the entire SQL-92 syntax at your finger tips, plus formatted printing and display of SQL scripts.
- Syntax-highlighted printing of SQL scripts.
- Command line options which allow you to integrate SQLScriptWriter into the Windows desktop.
- Developer conveniences such as auto-indent, editing history, and comment-out and parenthesise buttons.
- Context-sensitive help on SQL syntax (requires external help file)

SQLScriptWriter is a shareware product. For information on registration, and on how to contact the author, please see the relevant sections of this help file.

Working With Databases

SQLScriptWriter uses the Borland Database Engine (BDE) to provide portable access to major RDBMS platforms. ***In order to use SQLScriptWriter, you must have the BDE pre-installed, as SQLScriptWriter does not currently ship with the engine.*** My reasoning here is that you are unlikely to want to install a new driver architecture simply to run my product! You will also need suitable drivers to access your database. If you are currently doing Borland Delphi 2.x development using the BDE, you should have all you need.

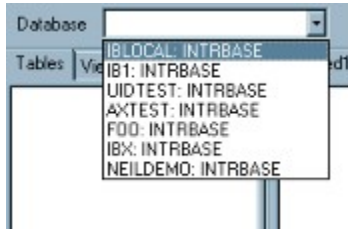
[Connecting To A Database](#)

[Refreshing The Database Connection](#)

[Disconnecting From A Database](#)

Connecting To A Database

SQLScriptWriter works most efficiently when you edit your scripts while connected to the database which they will be used on. To connect to a database, pull down the Database drop-down list box on the toolbar, and select the BDE alias for the database that you want to use.



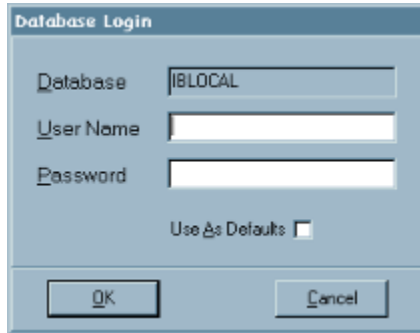
You will be presented with a login dialog: enter your user name and password, and press the OK button.

If you connected successfully, a "Connected" message will appear on the status bar, and the "Tables" and "Views" tabs in the metadata view will be populated with names of objects in the database.

If your connection failed, a failure message will appear on the status bar. This probably means your user name an/or password were not entered correctly. pull down the database list and try again.

Logging On To A Database

To log on to a database, you must first connect to it. Connection will cause the following log on dialog to be displayed.



The image shows a 'Database Login' dialog box with a blue title bar. It contains three text input fields: 'Database' with the text 'IBLOCAL', 'User Name', and 'Password'. Below the fields is a checkbox labeled 'Use As Defaults' which is currently unchecked. At the bottom of the dialog are two buttons: 'OK' and 'Cancel'.

To login, enter the user name and password, then press the OK button. If you are going to be logging on to the same database repeatedly during script development, you can make SQLScriptWriter remember your user name and password by checking the **Use As Defaults** checkbox.

Note: Default user name and password will be saved, unencrypted, in the SSW.INI file, which may cause security problems in production databases.

Disconnecting From A Database



To disconnect from a database, either press the **Disconnect** toolbar button, or choose **Disconnect** from the **Database** menu . If you want to connect to another database, you don't need to disconnect, simply choose the database you want from the toolbar drop-down list.

Refreshing The Database Connection



If you have made changes to your database metadata, for example by adding or dropping a view, the changes may not become visible immediately in the metadata display. To make sure the display is synchronised with the database state, press the **Refresh** button on the toolbar, or choose **Refresh** from the **Database** menu.

Text Editing

SQLScriptWriter allows you to edit ASCII text files using a tabbed text editor, which provides standard Windows methods for opening and saving files

[Creating A New File](#)

[Opening Files](#)

[Saving Files](#)

SQLScriptWriter uses standard Windows hot-keys for cut-and-paste, and for search and replace operations. It also provides the following control-key shortcuts:

[Ctrl-A](#) Toggles keyword completion

[Ctrl-B](#) Parenthesise (bracket) current selection

[Ctrl-G](#) Comment-out current selection

[Ctrl-T](#) Delete next word

[Ctrl-Y](#) Delete current line

[Ctrl-Q](#) Force keyword completion

[Ctrl-E](#) Execute selected text

To enhance your script writing, SQLScriptWriter provides the following editing conveniences:

[Auto-Indentation](#)

[Macros](#)

[Point & Click Use Of Metadata](#)

[SQL Keyword Completion](#)

If you want to make changes to the way the editor works, use the [Editor Settings](#) dialog.

Creating A New Script



To create a new script, select **New** from the **File** menu, or press the **New Script** button on the toolbar. This will create a new tab in the editor, but the file will not be given a name at this point.

Non-registered Users: When you create a new file, a line reminding you to register will be inserted at the beginning of the file. This will no longer occur if you [register SQLScriptWriter](#).

Opening Files



To open a file in SQLScriptWriter select **Open** from the **File** menu or press the **File Open** button on the toolbar. You will be presented with a Windows common dialog, which will allow you to navigate to the file you want to open. SQLScriptWriter defaults to listing script files (those with a .SQL extension), but you can select Text or All Files in the dialog.

You can also re-open a file using **Reopen** from the **File** menu, which provides you with a list of the last five files you have opened or saved.

If you specify a file which is already opened, the editor will simply make that file's tab active, and will not re-load the file.

Non-registered Users: When you open a file, a line reminding you to register will be inserted at the beginning of the file. This will no longer occur if you register SQLScriptWriter.

Saving Files

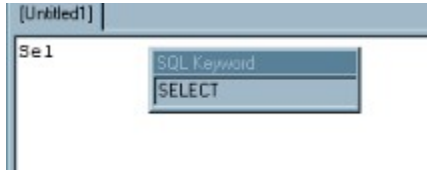


To save a file in SQLScriptWriter, either use **Save** from the **File** menu, or press the **Save File** button. If it have not given your file a name, you will be presented with a standard Windows Save As dialog .

Non-registered Users: When you save a file, a line reminding you to register will be inserted at the beginning of the file. This will no longer occur if you [register SQLScriptWriter](#).

Keyword Completion

SQLScriptWriter knows the entire set of SQL-92 reserved words. Using this knowledge, SQLScriptWriter can help you to write your scripts by providing keyword completion. This means that you only need to type in the first, distinguishing characters of a SQL keyword, and then let SQLScriptWriter provide the rest.



Keyword completion is enabled by default, so to try it out simply start to type in a SQL keyword such as 'SELECT'. If you have the floating keyword window open, you should see that when you type the initial 'S', the keyword 'SCHEMA' is displayed, as this is the first keyword starting with 'S'. Then, when you type 'E', the keyword 'SECOND' is displayed. Finally, you type 'L' and 'SELECT' appears.

To insert the completed keyword into the text, simply press SPACE, TAB or ENTER. Notice that the case of the keyword may be changed from what you actually typed; this is governed by the [SQL Text Options](#) dialog.

You can turn space-bar completion off, using the [Editor Settings dialog](#). If you do turn it off, you can still use it on occasion, by typing **Ctrl-Q** instead of SPACE. You can also toggle completion temporarily using **Ctrl-A**.

Toggling Keyword Completion

Keyword completion is very useful, but can be annoying if you are entering non-SQL text, or a comment in your script. You can turn it off altogether using the Editor Settings from the **Options** menu, or toggle it on/off by typing Ctrl-A.

Auto-Indentation

SQLScriptWriter provides auto-indentation, which means that the start of each new line is automatically aligned with the start of the previous one. To un-indent a line, use the BACKSPACE key. Auto-indentation can be turned on and off using the **Editor Settings** dialog from the **Options** menu.

Deleting Words and Lines

SQLScriptWriter provides the classic WordStar keys **Ctrl-T** and **Ctrl-Y** to delete the next word and the current line, respectively.

Commenting-Out Text



SQLScriptWriter provides the ability to comment-out selected text using SQL `/* ... */` comment delimiters. Simply select the text, and press the **Comment Out** button on the toolbar. You can use the same button, or its shortcut (Ctrl-G) to create a new comment if no text is selected.

Parenthesising Text



To parenthesise selected text, click on the **Parenthesise** button on the toolbar. If no text is selected, a pair of parens will be inserted in the text you are editing.

Context-sensitive SQL Help

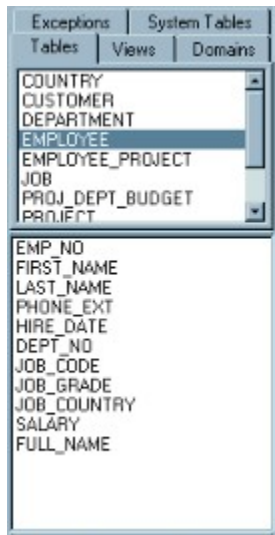


SQLScriptWriter allows you to make use of your RDBMS's existing SQL help files. Pressing the **SQL Help** button (or using the **SQL** option in the **Help** menu) will provide context-sensitive help on a keyword selected in the editor, or the help contents if there is no selection. For this to work, you must configure an external help file.

Using Metadata

Metadata is data which describes the structure of your database. An example of metadata is a list of all of the table names in the database.

One of the problems of writing SQL scripts is remembering the names of all the fields, tables, views, domains and other database objects. SQLScriptWriter helps you by presenting the names of these database objects in an easily accessible list format.



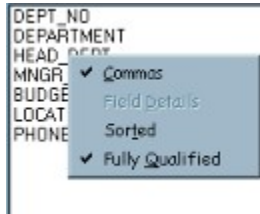
The database metadata is displayed in a split window to the left of the main editing pane. The top half provides tabbed lists of various kinds of metadata; exactly what appears here will depend on the driver for the database you are connected to, but it will always include displays for Tables and Views. You can customise the other tabs to suit the RDBMS you use; some customisations are supplied for Borland InterBase.

The bottom half of the pane will display database fields, but only if the top pane displays a table or view.

You can incorporate metadata text into your script in one of two ways; by using point-and-click selection, and by using macros.

Metadata Point-And-Click

SQLScriptWriter allows you to copy metadata information into your script simply by double-clicking on the metadata display. The name will be inserted at the current cursor position in the active editor tab, using the case settings from the SQL Text dialog. You can modify the way that SQLScriptWriter inserts the metadata text using a right-click context menu.



For fields, the context menu controls the following:

- Commas: If ticked, a comma is appended to the database object name.
- Field Details: If ticked, the field types (i.e. VARCHAR(32)) are displayed.
- Sorted: If ticked, the fields are sorted alphabetically.
- Fully Qualified: If ticked, fields will be preceded by table names, when inserted into the editor

Tables are slightly different (they always display in sorted order):

- System tables: If ticked, system tables are displayed
- Commas: If ticked, a comma is appended to the database object name.

Registering SQLScriptWriter

SQLScriptWriter is shareware. If, after your trial period of 28 days, you find it useful you should register it. Registration gives you the following benefits:

- All **nagware features of the product will be removed**, particularly the splash-screen, and the banner inserted into every script and print-out.
- You will be informed of any updates to the product, and will be entitled to a **free upgrade** to the next major release.
- You will get a **guaranteed email response** from me in case of any problems with the product.

Individual registrations cost **\$20.00**. For details of prices for multiple licences, please [contact me via email](#).

You can register SQLScriptWriter in the following ways

[Via CompuServe](#)
[By using a credit card.](#)

Once you have registered, please [email me](#), stating what name (max 32 characters) you would like to be registered as. When I receive notification of your registration, I'll email you your registration key (this will normally be within 24 hours of your registration). You can then use the SQLScriptWriter [registration dialog](#) to register your copy.

Once registered, two new entries will be made in your [SSW.INI](#) file. These are:

```
UserId=Your Name  
RegKey=Your Key
```

If the file gets damaged, or you decide to un-install and then re-install SQLScriptWriter, you can add these lines manually, or go through the registration dialog again. In any event, make sure you hang on to your registered name and key!

Registration Dialog

Once you have your registration key for your copy of SQLScriptWriter, select the **Register...** option from the **Help** menu. This will pop up the following dialog:



Type in your **User Name**, which should be exactly the same (in both case and spacing) as the name you used to obtain your key. Then type in the **Key**, once again being careful to make sure case and spacing match that of the key you were supplied with. Once you're happy with these, press the **Register Now** button.

If all is OK, you should be presented with the following confirmation:



If registration is not successful, re-check that you have typed your name and key exactly as they appear in the email registration notification you received. Note that after a few registration attempts, SQLScriptWriter will not let you try again. If you honestly need to make more attempts, shut down SQLScriptWriter and re-start before trying again.

If problems with registration persist, please [contact me](#) .

Contacting The Author

If you have any comments or questions regarding SQLScriptWriter, please contact **Neil Butterworth** by email at

Compuserve: 101565,2326

Internet: 101565.2326@compuserve.com

My Web home page, where news of new versions of SQLScriptWriter can be found, is at

http://ourworld.compuserve.com/homepages/neil_butterworth

I look forward to hearing from you!

Registration Via CompuServe

If you are a CompuServe member, the easiest way to register SQLScriptWriter is through CompuServe Shareware Registration. The **\$20.00** registration fee will be added to your normal CompuServe charges.

To register SQLScriptWriter, **GO SWREG** and then choose Register Shareware. The registration ID for SQLScriptWriter is **14166**.

Registration By Credit Card

Credit card registration of SQLScriptWriter, costing **\$20.00**, can be obtained via NorthStar Solutions, who will process orders you may wish to place using your Visa, MasterCard or Discover card. NorthStar Solutions can be easily contacted **for orders only** via any of the following methods:

Phone Orders

Available 10am - 8pm EST, Monday - Saturday
1-800-699-6395 (from US only)
1-803-699-6395

Fax Orders

1-803-699-5465 (available 24hrs)

Internet Orders

Fill in the online order form at <http://ourworld.compuserve.com/homepages/starmail>

Email Orders

AOL: STARMAIL
CompuServe 71561,2751
Internet: 71561.2751@compuserve.com

Please provide, or be prepared to provide:

The name of the program you are registering: **SQLScriptWriter**.

The product ID: **1177**

Your mailing address

Your credit card number and expiration date

Your email address

IMPORTANT: NorthStar Solutions process credit card registrations only. They cannot provide technical assistance. Please contact me with any problems regarding SQLScriptWriter.

Executing Your Script

Once you have entered some SQL into one of the SQLScriptWriter edit tabs, you can execute it against your database.



The **Execute Text** button will execute the first statement (terminated by a semi-colon) in any text you have selected in the current edit pane. You must be connected to a database in order to execute SQL in this way. If the executed statement is a SELECT, the result set will be displayed in the data view pane. Executing SQL in this way causes an automatic COMMIT to be performed if the statement was successful.

You can also execute selected text by using the **Ctrl-E** keyboard shortcut.

Once text has been successfully executed, it will be added to the SQLScriptWriter History View .



The **Execute Script** button will execute the entire script in the current edit tab. If you have not saved your script, SQLScriptWriter will ask you to do so. Executing SQL in this way does not display result sets. Instead, the SQL being executed is echoed in the status view pane, along with any error messages that may be produced.

If an error occurs when executing an entire script, the changes it would have made are rolled back, in so far as a particular RDBMS may allow. It is up to you to place COMMIT statements in your script to commit any changes to the database.

By default, scripts are executed against the currently connected database. However, you can also hard-code database connection information into your script, using the CONNECT pseudo-statement.

Both of these methods of executing SQL can also be performed via the **Execute** submenu of the **Database** menu.

The CONNECT Statement

The CONNECT statement as implemented by SQLScriptWriter is not a true SQL statement, but does allow you to hard-code database connection information into your script. It looks like this:

```
CONNECT "dbalias" USER "userid" PASSWORD "pword";
```

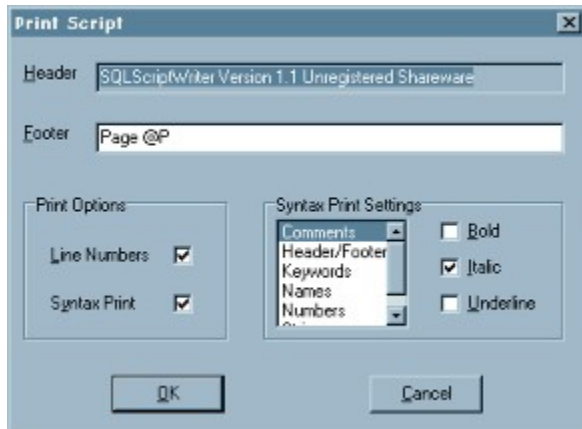
Where dbalias is a BDE alias for your database, userid is the id of a user, who should have sufficient privileges to execute the script successfully, and pword is their password. An example of a CONNECT might be:

```
CONNECT "iblocal" USER "NeilB" PASSWORD "mxyzptlk";
```

Printing



SQLScriptWriter provides syntax-highlighting for printed SQL scripts. To print the script in the current edit tab, press the **Print** button, or select **Print...** from the **File** menu. The following dialog will be displayed.



You can specify the **Header** (registered copies only) and **Footer** you want to appear on each page. Dates, times, filenames and page numbers can be included using a special syntax:

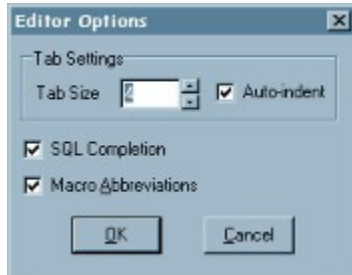
- @P print current page number
- @F print current full filename
- @D print date
- @T print time
- @@ print @ symbol

You can specify whether you want line numbers, and whether syntax highlighting should be performed via the **Print Options**. If highlighting is selected, use the **Syntax Print Settings** to specify how you would like each element to be printed.

Normal Windows printer setup is available via the **Print Setup...** option of the **File** menu.

Editor Settings

To change the way the editor works select **Editor Settings** from the **Options** menu, which will display this dialog:



The **Tab Settings** allow you to specify the character width of a tab and whether auto-indent should be used. **SQL Completion** governs whether the SQL Keyword Completion feature is on or not (you can also temporarily enable or disable keyword completion) . The **Macro Abbreviation** check-box governs whether abbreviations are expanded into their corresponding macro.

Macros

SQLScriptWriter provides you with macros with which allow you to automate production of boilerplate SQL code. Some macros are provided as part of the SQLScriptWriter distribution, and you can modify these or create your own.

The macro display consists of a number of tabbed lists, which are located beneath the metadata display area.



You can insert a macro's expansion into your text by double-clicking, or by using its abbreviation (if it has one). Abbreviations appear in the metadata display after the macro name. In the example above, the macro CreateTable has the abbreviation CT.

Out of the box, SQLScriptWriter provides four macro tabs, but you can add to these by customising your MACROS.INI file. The tabs provided by default are:

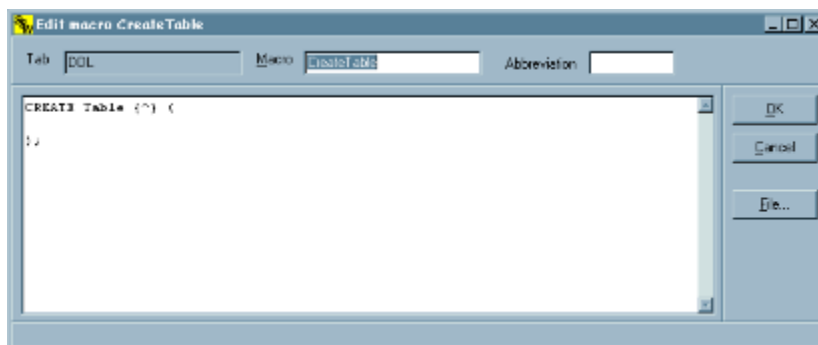
Shortcuts which gives you fast access to common SQL "phrases". For example, the SelectStar macro expands to SELECT * FROM.

DML which provides some Data Manipulation Language macros, including the ability to build a select statement from the current table in the metadata view .

DDL which provides Data Definition macros, such as the ability to create a view from a table.

Text which provides non-SQL textual macros.

To create new macros, use the **New...** option from the **Macros** menu. This will present you with the following dialog, which allows you to add a new macro to the current macro tab (the same dialog can be used to edit existing macros):



Give the macro a name, and if you wish an abbreviation, then enter the text which will be inserted in your

code. If the macro is very lengthy, you can save it in a separate text file, and then use the File... button to specify the path to the file, which will look something like this (notice the @ sign):

```
@C:\Usr\Sqledit\Macros\comment.mac
```

Macros can use special symbols, which will be expanded when the macro is executed. The symbols currently available are:

{database}	expands to name of database e.g. IBLOCAL
{dbtype}	expands to name of driver e.g. INTRBASE
{user}	expands to id of current user
{password}	expands to password of current user
{date}	expands to today's date
{time}	expands to current time
{ <u>object</u> }	expands to name of current database object
{ <u>fields</u> }	expands to name(s) of current fields, if any
{parameters}	as for {fields}, but preceded by a colon character
{ <u>setclause</u> }	expands to a number of SET clauses
{ <u>valuelist</u> }	expands to a list made up of current row in the data pane
{^}	positions cursor when macro terminates

Note that if a special symbol cannot be expanded it is removed. This may happen, for example, if a macro containing the symbol {object} is used when no object is selected in the metadata view.

Abbreviations

SQLScriptWriter supports the use of abbreviations for macros. Instead of double-clicking the macro name, you can type the abbreviation into the editing window, and macro expansion will happen automatically.

As an example of this, type NN into an editing window and then press SPACE, TAB or RETURN. You should see that NN was expanded into NOT NULL (if this doesn't work, you have probably edited the standard MACROS.INI that ships with SQLScriptWriter).

You can attach an abbreviation to any macro using the macro editor dialog. Abbreviations appear in the macro pane after the macro they abbreviate. They can contain any characters except SPACE, TAB, NEWLINE and =. In general, the macros supplied with SQLScriptWriter only have abbreviations for Shortcut and Text macros, but this is not enforced by the package. If you don't want macro abbreviations to occur at all, turn them off using the Editor Settings dialog.

History View

Once a SQL command has been successfully executed, it is added to the SQLScriptWriter **History View**, from where it can be repeatedly executed by double-clicking. The history view is switched off by default; to enable it, choose **History** from the **View** menu.

Note that SQL commands executed as part of script execution are not added to the history view.

Special Macro Symbols

Some macro special symbols work in conjunction with the metadata and data panes.

{object} will insert the name of the object currently selected in the upper, tabbed metadata display, if there is one.

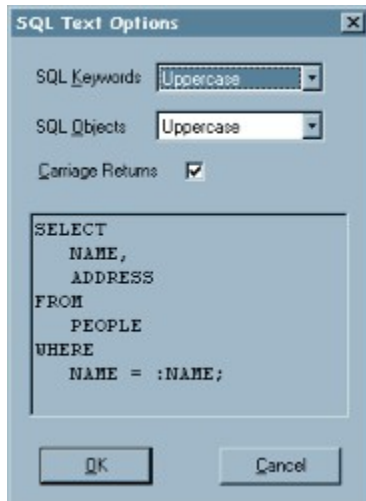
{fields} will insert the currently selected names in the lower metadata pane. If none are selected, then *all* are inserted. This is especially useful for writing views, select statements and creating tables.

{setclause} inserts a SQL SET clause of the form **SET name =** , In other words, it works like {fields} does, but inserts SET in front of the field name and = afterwards. This can be used to write UPDATE statements.

{valuelist} inserts a comma-separated, quoted list of the values in the currently selected row in the data view. This is useful for writing INSERT scripts.

SQL Text Options

The SQL Text Options dialog governs how completed SQL keywords, and database object names (e.g. table and field names) are displayed in the editor.



In this version of SQLScriptWriter, SQL Text options are not applied to macro text.

Configuring External Help File

SQLScriptWriter enables you to use the Windows SQL help file that goes with your RDBMS (assuming there is one). Correctly configured, you can use an external help file to provide context-sensitive help on SQL syntax.

To configure your help file, you need to add a line to the [Settings] section in your SSW.INI file. The line should be of the form

```
ExternalHelpFile=Help file path and name
```

For example, to use Borland's InterBase help file, you might enter something like this:

```
ExternalHelpFile=D:\IntrBase\BIN\SQLREF32.HLP
```

Of course, the path and file name will vary between installations and RDBMSs.

The SSW.INI File

SQLScriptWriter is configured via two .INI files: SSW.INI for the main program, and MACROS.INI for macro texts. These .INI files will be located within the directory you installed SQLScriptWriter into. SQLScriptWriter makes no changes to the Windows registry, and installs no files in the Windows system directories.

In general, you should not need to edit either .INI file directly. The only exceptions to this are when you want to create a new macro tab, when you configure an external help file, and when you customise your metadata display for a specific RDBMS. However, an example SSW.INI file is presented here so you aren't in the dark about what's going on!

```
; Sample SQLScriptWriter INI File
```

```
[INTRBASE.ADDONS]
; see customising your metadata for details of this section
Domains=SELECT RDB$FIELD_NAME FROM RDB$FIELDS WHERE NOT RDB$FIELD_NAME LIKE
'RDB$%'
Exceptions=SELECT RDB$EXCEPTION_NAME FROM RDB$EXCEPTIONS WHERE NOT
RDB$EXCEPTION_NAME LIKE 'RDB$%'
System Tables=SELECT RDB$RELATION_NAME FROM RDB$RELATIONS WHERE
RDB$RELATION_NAME LIKE 'RDB$%'
```

```
[Settings]
;these store the SQL Text options
AddCommaToDetail=True
AddCommaToObject=True
SQLKeywordCase=Uppercase
SQLObjectCase=Uppercase
CRInSQLList=True
FullyQualifiedFieldNames=False;

; this specifies an external help file
ExternalHelpFile=D:\IntrBase\BIN\SQLREF32.HLP
```

```
;this controls keyword completion
KeywordCompletion=True
```

```
;these are Editor Settings
TabSize=4
AutoIndent=True
KeywordCompletionDisplay=True
BeepOnError=True
```

```
;these are most recently used scripts
LastFile0=C:\Usr\Sqledit\makedb.SQL
LastFile1=C:\Usr\Sqledit\makeview.sql
LastFile2=C:\Usr\Sqledit\dropview.sql
LastFile3=C:\Usr\Sqledit\fixup.sql
LastFile4=C:\Usr\Sqledit\cc.SQL
```

```
;these are Printer Options
PrintFooter=Page @P
PrintHeader=SQLScriptWriter Version 1.2 Registered Copy
PrintLineNumbers=True
```

PrintSyntax=True

; This is your registration key (registered only)
RegKey=XXXXX

; This is your registered user name (registered only)
UserId=Mr Nice Guy

; Default uid/password from login dialog
DefaultDBPassword=masterkey
DefaultDBUID=sysdba

; Controls macro abbreviations
Abbreviations=True

; These control syntax printing
SPComments=I
SPHeader/Footer=BI
SPKeywords=B
SPNames=
SPNumbers=
SPStrings=

The MACROS.INI File

The SQLScriptWriter macro pane(s) and contents are configured via the file MACROS.INI. If you produce many macros, you may wish to modify this file directly, so the format is described here. In general though, you should be able to use the Macro Editing dialog for most of your requirements.

MACROS.INI consists of a single section [Macro.Tabs] describing the tabs in the macro pane, plus a number of sections describing each pane. The tabs section looks something like this:

```
[Macro.Tabs]
Clauses=Clauses
DDL=DDL
DML=DML
DriverIntrbase=Interbase
Text=Text
```

Each section entry will cause a tab to be created in the macro pane at program startup except for any entries whose name begins with "Driver". These will only be created if the driver for the current database matches the sub-string following "Driver". In this case, the tab "DriverIntrbase" will only be created if a database using a driver called "Intrbase" (the Borland InterBase driver, as it happens) is accessed.

The title of each tab is string on the right of the = sign, so in this example the DriverIntrBase tag actually has the visual title "Interbase".

For each entry in the Macro.Tabs section, there is a section in the file. An example of the DML section might be:

```
[Shortcuts]
SelectStar SS="SELECT * FROM"
InnerJoin IJ="INNER JOIN"
OuterJoin OJ="OUTER JOIN"
NaturalJoin NJ="NATURAL JOIN"
```

The section title must be the same as a name (left hand side of = sign) in the Macro.Tabs section. The name of the macro, which appears in the macro pane, is a name in this DML section. The text of the macro is the value. Special characters can be used for tabs (t) and newlines (n), but note these are not required if you use the Macro Editor dialog.

If your macro is very long (for example, a standard file header with copyright and other messages), you can place it in a separate file. Use the '@' character to dereference the file within the macro.

```
StandardHeader=@c:\docs\legal\stdhdr.mac
```

If you have created abbreviations for any of your macros, these will appear under a special section called (suprisingly) [Abbreviations]. It should look something like this:

```
[Abbreviations]
SS=Shortcuts:SelectStar SS
IJ=Shortcuts:InnerJoin IJ
OJ=Shortcuts:OuterJoin OJ
NJ=Shortcuts:NaturalJoin NJ
```

You are recommended not to edit this section manually.

Customising SQLScriptWriter

You can customise SQLScriptWriter to match your working methods and environment in the following ways:

[Change print headers, footers and syntax highlighting](#)

[Change editor settings](#)

[Specify SQL text options](#)

[Create new macros](#)

[Add customised metadata lists](#)

[Use command line switches to integrate with Windows desktop](#)

[Specify an external SQL help file](#)

See also the information on the [SSW.INI](#) and [MACROS.INI](#) files.

Customising Metadata Display

SQLScriptWriter allows you to add customised tabs to the [metadata display](#) . This facility allows you to customise SQLScriptWriter to work with your specific RDBMS. SQLScriptWriter ships with customisations for Borland InterBase which provide Domain, Exception and System Table views. In order to understand how to perform your own customisation, it's a good idea to examine how these have been provided.

The [SSW.INI](#) file contains a section which looks like this:

```
[INTRBASE.ADDONS]
; Domains tab
Domains=SELECT RDB$FIELD_NAME FROM RDB$FIELDS WHERE NOT RDB$FIELD_NAME LIKE
'RDB$%'

; Exceptions tab
Exceptions=SELECT RDB$EXCEPTION_NAME FROM RDB$EXCEPTIONS WHERE NOT
RDB$EXCEPTION_NAME LIKE 'RDB$%'

; System tables tab
System Tables=SELECT RDB$RELATION_NAME FROM RDB$RELATIONS WHERE
RDB$RELATION_NAME LIKE 'RDB$%'
```

You can see that the section name [INTRBASE.ADDONS] consists of the database driver name (INTRBASE) a period, and word ADDONS. If you wanted to create a customisation for another RDBMS, called for example SKYBASE, you would create a section in the .INI file called [SKYBASE.ADDONS].

Creating the tabs themselves requires you to know something about the databases system tables. Let's assume that SKYBASE keeps its Domain names in a table called SYS_DOMAINS, where the column name is DOM_NAME. To create a tab for SKYBASE domains, your .INI file section should look like this:

```
[SKYBASE.ADDONS]
; domains from SKYBASE
Domains=SELECT DOM_NAME FROM SYS_DOMAINS;
```

The name on the left hand side of the = sign is the name that will appear on the tab. The string on the right hand sign must be a SELECT statement, which should return a single column.

Once you've added a customisation, whenever you use a SKYBASE database, the tabs will appear and be populated. Note that in this release of SQLScriptWriter you cannot populate the detail pane of the metadata display.

Data View Pane

The data view provides a window on the currently selected table or view in the metadata pane.

EMP_NO	FIRST_NAME	LAST_NAME	PHONE_EXT	HIRE_DATE	DEPT_NO	JOB_CODE	JOB_L
2	Robert	Nelson	250	28/12/88	600	VP	
4	Bruce	Young	233	28/12/88	621	Eng	
5	Kim	Lambert	22	06/02/89	130	Eng	
8	Leslie	Johnson	410	05/04/89	180	Mktg	
9	Phil	Forest	229	17/04/89	622	Mngt	
11	K. J.	Weston	34	17/01/90	130	SRep	
12	Teri	Lee	256	01/05/90	000	Admin	

This window is read-only, SQLScriptWriter does not support editing of database tables. However, the window is very useful when developing views and other scripts, as a means of displaying what the view would produce.

The data view is also used for displaying the result of the last individual SELECT statement executed. If a script is being executed, then the data view is not used, but instead is replaced by the status view, which displays each statement as it is executed.

One powerful feature of the data view is the ability to examine text BLOBs using the **memo view**. This view allows you to single-click on any memo (text BLOB) field within the data view and display its contents. To pop up the memo view window, select **Memo** from the **View** menu.

LE	MIN			
ant				
ative Assistant				
ative Assistant				
ative Assistant	20000	40000	(Memo)	(Bytes)
icutive Officer	130000	250000	(Memo)	(Bytes)
ancial Officer	85000	140000	(Memo)	(Bytes)
	75000	120000	(Memo)	(Bytes)

Memo View: JOB_REQUIREMENT

CPA with 3-5 years experience.
Spreadsheet, data entry, and word processing knowledge required.

Execution Status Pane

The execution status pane replaces the data view pane whenever you execute an entire script from within SQLScriptWriter. The status pane will display each SQL statement as it is executed, together with any error messages.

Status Bar

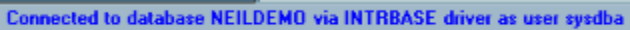
The status bar, at the bottom of SQLScriptWriter's window, keeps you informed of what is going on. It's got three states:

Normal - nothing is happening



SQLScriptWriter Version 1.2

Everything is working fine



Connected to database NEILDEMO via INTRBASE driver as user sysdba

Houston, we have a problem



Could not connect to NEILDEMO. User name or password may be incorrect

After a few seconds, the status bar will always return to the "normal" state. If you missed the status/error message it was displaying, you can recall it by clicking on the status bar.

Command Line Switches

SQLScriptWriter supports command line switches which allow you to configure Windows desktop context menus. The three switches supported are

-E scriptname	which edits the script
-X scriptname	which executes the script
-P scriptname	which prints the script

For example, the following command will open for editing the script called MAKEVIEW.SQL

```
SSW -E MAKEVIEW.SQL
```

This ability allows you to use SQLScriptWriter as a server for right-click context menus. Here's a screenshot of Microsoft Explorer about to call SQLScriptWriter to print a .SQL file.



To setup the right-button context menus to work in this way, perform the following steps:

- From the Explorer **View** menu, choose **Options...** This will pop up the **Options Dialog**.
- Select the **File Types** tab, and press the **New Type...** button.
- Fill in the **Description Of Type** (for example, SQL Script) and the **Associated Extension** (for example SQL), then press the **New..** button. This will present another dialog.
- Fill in **Action** (for example, Print), and Application used to perform action. In this field, you must enter the path to the SQLScriptWriter executable and the command line. For example:
- `\DBMS\BIN\SSW\SSW.EXE -P "%1"`
- The important things here are getting the path right (there's a Browse... button), using the right switch (-P for print) and the "%1" (don't forget the quotes).

Repeat for the -E and -X options as desired.

Getting Started

Here's a short tour of some of SQLScriptWriter's features. Make sure that you have SQLScriptWriter fired up, and that there is at least one database that you can connect to.

First, connect to your database and then login to it.

Now, look at the metadata pane. It should be displaying a list of tables in the database.

Click on one of the tables. Notice that the lower half of the metadata pane gives you a list of columns in the table, and that the data view pane is filled with the actual data contained in the table.

Double-click on one of the column names. Notice that this pastes the column name into the currently active edit.

With the mouse, select several column names (use Control-Left Button). Now go to the "DML" tab of the macro pane, and double-click on "Select". A SQL SELECT statement should be automatically generated for you!

In the edit pane, select the whole of the SELECT statement, and press the Execute Text button. This will execute your SELECT statement, and the results will be displayed in the data pane.

Open a new edit pane by pressing the New script button. Now start typing in a SELECT statement of your own. as soon as you've typed SEL, you should see that "SELECT" appears in the small keyword completion window. Press the spacebar to complete the keyword.

Once again in the edit pane, try out an abbreviation. Type in NN and hit SPACE and you should see that it has been expanded to NOT NULL. You can set up abbreviations and macros for all your common SQL constructs.

Try commenting out and bracketing the keyword, using the Comment out and Parenthesise buttons. You'll need to select the keyword first.

Finally, disconnect from the database by pressing the Disconnect button

