

sgNeoDatabase

Complete Continuous Help File for the Latest Build

VisualNEO Win Business Application Stack

Build basis: current supplied sgNeoDatabase.nbp and the matching latest source tree (1.1.3 deployment package).

Command coverage: 127 registered sgneo* actions across database, grid, chart, PDF, print, mail, crypto and form-builder functionality.

This manual is intentionally continuous: installation, architecture, quick-start recipes, every registered command, deployment notes, and troubleshooting are presented in one PDF.

Important: Examples are practical NeoScript patterns based on the current source registrations. Replace sample database/table names, credentials, paths and variables with values from your application.

Contents

Section	Commands
neoDatabase — Modern Database Engine	49
neoDataGrid — Editable Data Grid	29
neoChart — Interactive Charts	12
neoPDF — PDF Reports	3
neoPrint — Preview and Printing	3
neoMail — SMTP / IMAP / POP3	10
neoCrypto — Hashing, Encoding, AES and Passwords	18
Form Builder — Generated Data Entry Forms	3
Quick Start and Architecture	—
Deployment and DLLs	—
Practical Recipes	—
Error Handling and Compatibility	—

1. Quick Start

The basic VisualNEO workflow is: open a database, execute a query, optionally display the result in a grid or chart, then close the handle.

SQLite

Open a local SQLite database:

```
sgneoDatabaseOpen "[PubDir]library.db" "[DB]"
```

```
sgneoDatabaseQuery "[DB]" "SELECT * FROM books WHERE author=?" "[Tolkien]" "[Result]"
```

```
sgneoDatabaseClose "[DB]"
```

PostgreSQL

A typical connection string is:

```
sgneoDatabaseOpen "Driver=PostgreSQL;Server=127.0.0.1;Port=5432;Database=nbu_erp;User=postgres;Password=postgres" "[DB]"
```

For the supplied Win32 build, PostgreSQL client libraries must be Win32/x86. The current deployment package registers the x86 libpq/OpenSSL 1.1 dependency set when those files are present.

MySQL / MariaDB

Example connection:

```
sgneoDatabaseOpen "Driver=MySQL;Server=127.0.0.1;Database=nbu_erp;User=root;Password=pw" "[DB]"
```

Query parameters

The current database action descriptions support parameter lists represented as JSON arrays/objects or pipe-separated values. Both positional ? placeholders and named :name placeholders are supported by the documented parameter layer.

2. Architecture and Deployment

The latest build is a Win32 VisualNEO plug-in. SQLite is statically linked in the FireDAC layer, while PostgreSQL/MySQL/MariaDB and TLS functionality can depend on external client libraries.

Runtime dependency registration

The deployment unit registers only existing Win32 PE files. The current source explicitly handles:

Area	Registered Win32 dependency set
PostgreSQL	libpq.dll; libssl-1_1.dll; libcrypto-1_1.dll; libintl-8.dll; libiconv-2.dll; msvcrt120.dll; vcruntime140.dll
neoMail TLS	libssl-3.dll; libcrypto-3.dll
MySQL/MariaDB	libmariadb.dll; libmysql.dll
SQLite	No external sqlite3.dll required by the statically linked SQLite wrapper

Important distinction: nbAddFile registers external dependency files for VisualNEO publication packaging. It does not physically turn the DLLs into bytes inside the .nbp binary. The .nbp remains the plug-in DLL.

Win32 / x64 rule

Do not place x64 PostgreSQL/OpenSSL client DLLs beside this Win32 plug-in. The deployment helper checks the PE machine type and refuses non-I386 DLLs.

TLS mail

The supplied documentation specifies 32-bit OpenSSL 3 DLLs for TLS mail: libssl-3.dll and libcrypto-3.dll. Gmail examples use smtp.gmail.com:587 with STARTTLS and an App Password.

3. neoDatabase — Modern Database Engine

49 commands are registered in the latest source build for this component.

1. sgneoDatabaseOpen

Open a database (SQLite file, connection string or URI) and store its handle in a variable.

Parameter	Type
Source	Text / expression
[Var]	Result variable

Example

```
sgneoDatabaseOpen "[PubDir]library.db" "[DB]"
```

2. sgneoDatabaseOpenEx

Open a database with a separate options list (Password=...;Pooled=1;PoolMax=8;Encrypt=1).

Parameter	Type
Source	Text / expression
Options	Text / expression
[Var]	Result variable

Example

```
sgneoDatabaseOpenEx "Driver=MySQL;Server=127.0.0.1;Database=nbu_erp;User=root;Password=pw" "Pooled=1;PoolMax=8" "[DB]"
```

3. sgneoDatabaseClose

Close a database handle.

Parameter	Type
DB	Text / expression

Example

```
sgneoDatabaseClose "[DB]"
```

4. sgneoDatabaseCloseAll

Close every open database handle.

Parameter	Type
-----------	------

Example

```
sgneoDatabaseCloseAll
```

5. sgneoDatabaseQuery

Run a SELECT and return all rows as JSON (or CSV) into a variable. Params: JSON array/object or a|b|c.

Parameter	Type
DB	Text / expression
SQL	Text / expression
Params	Text / expression
[Var]	Result variable

Example

```
sgneoDatabaseQuery "[DB]" "SELECT id,title FROM books WHERE author=?" "[Tolkien]" "[Result]"
```

6. sgneoDatabaseQueryRow

Run a SELECT and return only the first row as a JSON object.

Parameter	Type
DB	Text / expression
SQL	Text / expression
Params	Text / expression
[Var]	Result variable

Example

```
sgneoDatabaseQueryRow "[DB]" "SELECT * FROM books WHERE id=?" "[BookId]" "[Row]"
```

7. sgneoDatabaseQueryValue

Run a SELECT and return the first column of the first row (scalar).

Parameter	Type
DB	Text / expression
SQL	Text / expression
Params	Text / expression
[Var]	Result variable

Example

```
sgneoDatabaseQueryValue "[DB]" "SELECT COUNT(*) FROM books" "" "[Count]"
```

8. sgneoDatabaseExecute

Run INSERT/UPDATE/DELETE/DDl with parameters; stores rows affected in a variable.

Parameter	Type
DB	Text / expression
SQL	Text / expression
Params	Text / expression
[Var]	Result variable

Example

```
sgneoDatabaseExecute "[DB]" "UPDATE books SET price=? WHERE id=?" "[Price|BookId]" "[Affected]"
```

9. sgneoDatabaseExecuteScript

Run several semicolon-separated SQL statements (schema creation, seed data).

Parameter	Type
DB	Text / expression
Script	Text / expression

Example

```
sgneoDatabaseExecuteScript "[DB]" "CREATE TABLE IF NOT EXISTS books(id INTEGER PRIMARY KEY,title TEXT)"
```

10. sgneoDatabaseBeginTransaction

Start a transaction.

Parameter	Type
DB	Text / expression

Example

```
sgneoDatabaseBeginTransaction "[DB]"
```

11. sgneoDatabaseCommit

Commit the current transaction.

Parameter	Type
DB	Text / expression

Example

```
sgneoDatabaseCommit "[DB]"
```

12. sgneoDatabaseRollback

Roll back the current transaction.

Parameter	Type
DB	Text / expression

Example

```
sgneoDatabaseRollback "[DB]"
```

13. sgneoDatabasePrepare

Prepare a statement once for repeated execution; stores a statement handle.

Parameter	Type
DB	Text / expression
SQL	Text / expression
[Var]	Result variable

Example

```
sgneoDatabasePrepare "[DB]" "UPDATE books SET price=? WHERE id=?" "[Stmt]"
```

14. sgneoDatabaseExecutePrepared

Execute a prepared statement with new parameters; stores rows affected.

Parameter	Type
Stmt	Text / expression
Params	Text / expression
[Var]	Result variable

Example

```
sgneoDatabaseExecutePrepared "[Stmt]" "[Price|BookId]" "[Affected]"
```

15. sgneoDatabaseQueryPrepared

Run a prepared SELECT with new parameters; stores the result.

Parameter	Type
Stmt	Text / expression
Params	Text / expression
[Var]	Result variable

Example

```
sgneoDatabaseQueryPrepared "[Stmt]" "[Tolkiens]" "[Result]"
```

16. sgneoDatabaseFinalize

Release a prepared statement handle.

Parameter	Type
Stmt	Text / expression

Example

```
sgneoDatabaseFinalize "[Stmt]"
```

17. sgneoDatabaseQueryAsync

Run a SELECT in the background; when done sets the variable and plays the callback action.

Parameter	Type
DB	Text / expression
SQL	Text / expression
Params	Text / expression
[Var]	Result variable
Subroutine	Text / expression

Example

```
sgneoDatabaseQueryAsync "[DB]" "SELECT * FROM books" "" "[Result]" "OnBooksLoaded"
```

Async note: the callback is a subroutine name. Use `sgneoDatabaseAsyncPending` when the application needs the number of background jobs still running.

18. sgneoDatabaseExecuteAsync

Run a non-SELECT statement in the background; sets rows affected and plays the callback.

Parameter	Type
DB	Text / expression
SQL	Text / expression
Params	Text / expression
[Var]	Result variable
Subroutine	Text / expression

Example

```
sgneoDatabaseExecuteAsync "[DB]" "UPDATE books SET price=price*1.05" "" "[Affected]" "OnUpdateDone"
```

Async note: the callback is a subroutine name. Use `sgneoDatabaseAsyncPending` when the application needs the number of background jobs still running.

19. sgneoDatabaseAsyncPending

Store the number of background jobs still running.

Parameter	Type
[Var]	Result variable

Example

```
sgneoDatabaseAsyncPending "[Pending]"
```

20. sgneoDatabaseLastInsertId

Store the last auto-generated id (PostgreSQL: optional sequence name).

Parameter	Type
DB	Text / expression
Sequence	Text / expression
[Var]	Result variable

Example

```
sgneoDatabaseLastInsertId "[DB]" "" "[NewID]"
```

PostgreSQL note: the Sequence parameter can be used when a specific sequence name is required.

21. sgneoDatabaseTables

List tables and views as JSON.

Parameter	Type
DB	Text / expression
[Var]	Result variable

Example

```
sgneoDatabaseTables "[DB]" "[Tables]"
```

22. sgneoDatabaseColumns

Describe the columns of a table as JSON.

Parameter	Type
DB	Text / expression
Table	Text / expression
[Var]	Result variable

Example

```
sgneoDatabaseColumns "[DB]" "books" "[Columns]"
```

23. sgneoDatabaseSchema

Full schema (tables with columns) as JSON.

Parameter	Type
DB	Text / expression
[Var]	Result variable

Example

```
sgneoDatabaseSchema "[DB]" "[Schema]"
```

24. sgneoDatabaseInfo

Connection information (driver, server version, state) as JSON.

Parameter	Type
DB	Text / expression
[Var]	Result variable

Example

```
sgneoDatabaseInfo "[DB]" "[Info]"
```

25. sgneoDatabaseTableExists

Store True/False depending on whether a table exists.

Parameter	Type
DB	Text / expression
Table	Text / expression
[Var]	Result variable

Example

```
sgneoDatabaseTableExists "[DB]" "books" "[Exists]"
```

26. sgneoDatabaseEncrypt

Encrypt an (unencrypted, closed) SQLite file with AES-256.

Parameter	Type
File	File name/path
Password	Text / expression

Example

```
sgneoDatabaseEncrypt "[PubDir]library.db" "secret"
```

27. sgneoDatabaseDecrypt

Remove encryption from a (closed) SQLite file.

Parameter	Type
File	File name/path
Password	Text / expression

Example

```
sgneoDatabaseDecrypt "[PubDir]library.db" "secret"
```

28. sgneoDatabaseChangePassword

Change the encryption password of a (closed) SQLite file.

Parameter	Type
File	File name/path
OldPassword	Text / expression
NewPassword	Text / expression

Example

```
sgneoDatabaseChangePassword "[PubDir]library.db" "oldpass" "newpass"
```

29. sgneoDatabaseIsEncrypted

Store the encryption state of a SQLite file (or algorithm).

Parameter	Type
File	File name/path
[Var]	Result variable

Example

```
sgneoDatabaseIsEncrypted "[PubDir]library.db" "[State]"
```

30. sgneoDatabaseBackup

Copy an open SQLite database to a new file (VACUUM INTO).

Parameter	Type
DB	Text / expression
TargetFile	File name/path

Example

```
sgneoDatabaseBackup "[DB]" "[PubDir]backup.db"
```

31. sgneoDatabaseSetOption

Set a plug-in option: ResultFormat (json|csv|tab), DateFormat, NullText, CSVDelimiter, UTF8, ShowErrors, IgnoreEncryption, PGVendorLib, MySQLVendorLib.

Parameter	Type
Name	Text / expression
Value	Text / expression

Example

```
sgneoDatabaseSetOption "ResultFormat" "json"
```

Documented options include ResultFormat (json|csv|tab), DateFormat, NullText, CSVDelimiter, UTF8, ShowErrors, IgnoreEncryption, PGVendorLib and MySQLVendorLib.

32. sgneoDatabaseJSONGet

Extract a value from JSON text by path, e.g. 0.title or count.

Parameter	Type
JSON	Text / expression
Path	Text / expression
[Var]	Result variable

Example

```
sgneoDatabaseJSONGet "[Result]" "0.title" "[FirstTitle]"
```

33. sgneoDatabaseDriverInfo

Report which database engines this build supports and whether their client libraries were found (JSON).

Parameter	Type
[Var]	Result variable

Example

```
sgneoDatabaseDriverInfo "[DriverInfo]"
```

34. sgneoDatabaseOpenTable

Generate VisualNeo text, Boolean and BLOB entry objects for a database table and place them on the clipboard for Paste onto the current page.

Parameter	Type
DB	Text / expression
Table	Text / expression
Prefix	Text / expression

Example

```
sgneoDatabaseOpenTable "[DB]" "applications" "APP"
```

This action generates VisualNEO table-form objects and places generated text on the clipboard for insertion into the current page. Follow the generated object/event instructions in the publication.

35. sgneoDatabaseFirst

Move the generated table form to the first record.

Parameter	Type
DB	Text / expression
Table	Text / expression

Example

```
sgneoDatabaseFirst "[DB]" "applications"
```

36. sgneoDatabasePrev

Move the generated table form to the previous record.

Parameter	Type
DB	Text / expression
Table	Text / expression

Example

```
sgneoDatabasePrev "[DB]" "applications"
```

37. sgneoDatabaseNext

Move the generated table form to the next record.

Parameter	Type
DB	Text / expression
Table	Text / expression

Example

```
sgneoDatabaseNext "[DB]" "applications"
```

38. sgneoDatabaseLast

Move the generated table form to the last record.

Parameter	Type
DB	Text / expression
Table	Text / expression

Example

```
sgneoDatabaseLast "[DB]" "applications"
```

39. sgneoDatabaseAddRecord

Append a record in the generated table form.

Parameter	Type
DB	Text / expression
Table	Text / expression

Example

```
sgneoDatabaseAddRecord "[DB]" "applications"
```

40. sgneoDatabaseSaveEdits

Write the generated table form variables back to the current record.

Parameter	Type
DB	Text / expression
Table	Text / expression

Example

```
sgneoDatabaseSaveEdits "[DB]" "applications"
```

41. sgneoDatabaseDeleteRecord

Delete the current record in the generated table form.

Parameter	Type
DB	Text / expression
Table	Text / expression

Example

```
sgneoDatabaseDeleteRecord "[DB]" "applications"
```

42. sgneoDatabaseCloseTable

Release the generated table form session.

Parameter	Type
DB	Text / expression
Table	Text / expression

Example

```
sgneoDatabaseCloseTable "[DB]" "applications"
```

43. sgneoDatabaseTablesList

Load database tables/views into a VisualNEO ListBox or ComboBox and optionally store the same table metadata as JSON.

Parameter	Type
DB	Text / expression
List/Combo object	VisualNEO object name
[Var]	Result variable

Example

```
sgneoDatabaseTablesList "[DB]" "TableList" "[Tables]"
```

44. sgneoDatabaseSchemaRender

Render the database schema JSON as an interactive hierarchical tree inside a Rectangle or Container/Modern Container.

Parameter	Type
SchemaVar	Result variable
Rectangle/Container	VisualNEO object name
[Var]	Result variable
[Var]	Result variable

Example

```
sgneoDatabaseSchemaRender "[Schema]" "SchemaRect" "[Result]" "[Status]"
```

45. sgneoDatabaseSchemaDestroy

Destroy all active schema renderers and release their host windows. Use before changing or deleting schema host objects.

Parameter	Type
-----------	------

Example

```
sgneoDatabaseSchemaDestroy
```

46. sgneoDatabaseColumnsRender

Retrieve table column metadata as JSON and render it inside a VisualNEO Rectangle or Container.

Parameter	Type
DB	Text / expression
Table	Text / expression
Rectangle/Container	VisualNEO object name
[Var]	Result variable

Example

```
sgneoDatabaseColumnsRender "[DB]" "applications" "ColumnsRect" "[Result]"
```

47. sgneoDatabaseColumnsDestroy

Destroy all active column metadata renderers.

Parameter	Type
-----------	------

Example

```
sgneoDatabaseColumnsDestroy
```

48. sgneoDatabaseTablesRender

Retrieve database tables/views as JSON and render them inside a VisualNEO Rectangle or Container.

Parameter	Type
DB	Text / expression
Rectangle/Container	VisualNEO object name
[Var]	Result variable

Example

```
sgneoDatabaseTablesRender "[DB]" "TablesRect" "[Result]"
```

49. sgneoDatabaseTablesDestroy

Destroy all active table metadata renderers.

Parameter	Type
-----------	------

Example

```
sgneoDatabaseTablesDestroy
```

4. neoDataGrid — Editable Data Grid

29 commands are registered in the latest source build for this component.

1. sgneoDataGridCreate

Create a data grid inside an existing Rectangle object. Choose the Rectangle, then store the grid handle.

Parameter	Type
DB	Text / expression
RectangleName	VisualNEO object name
[Var]	Result variable

Example

```
sgneoDataGridCreate "[DB]" "Grid1" "[G]"
```

2. sgneoDataGridLoad

Run a SELECT and fill the grid. Parameters as in neoDatabaseQuery.

Parameter	Type
G	Text / expression
SQL	Text / expression
Params	Text / expression

Example

```
sgneoDataGridLoad "[G]" "SELECT * FROM books" ""
```

3. sgneoDataGridSetColumns

Set column captions/widths/order: "field=Caption:width;field2=Caption2:width2".

Parameter	Type
G	Text / expression
Columns	Text / expression

Example

```
sgneoDataGridSetColumns "[G]" "title=Title:200;author=Author:160"
```

4. sgneoDataGridSort

Sort the grid by a field. Direction is ASC or DESC.

Parameter	Type
G	Text / expression
Field	Text / expression
ASC/DESC	Text / expression

Example

```
sgneoDataGridSort "[G]" "title" "ASC"
```

5. sgneoDataGridRefresh

Re-run the grid's last query.

Parameter	Type
G	Text / expression

Example

```
sgneoDataGridRefresh "[G]"
```

6. sgneoDataGridGetSelected

Store the selected row as a JSON object.

Parameter	Type
G	Text / expression
[Var]	Result variable

Example

```
sgneoDataGridGetSelected "[G]" "[Row]"
```

7. sgneoDataGridOnSelect

When a row is clicked, set a variable to the row JSON and play a callback action.

Parameter	Type
G	Text / expression
[RowVar]	Result variable
Subroutine	Text / expression

Example

```
sgneoDataGridOnSelect "[G]" "[Row]" "OnRowPicked"
```

8. sgneoDataGridRowCount

Store the number of rows currently shown.

Parameter	Type
G	Text / expression
[Var]	Result variable

Example

```
sgneoDataGridRowCount "[G]" "[Rows]"
```

9. sgneoDataGridShow

Show the grid.

Parameter	Type
G	Text / expression

Example

```
sgneoDataGridShow "[G]"
```

10. sgneoDataGridHide

Hide the grid.

Parameter	Type
G	Text / expression

Example

```
sgneoDataGridHide "[G]"
```

11. sgneoDataGridMove

Reposition/resize the grid: "x,y,width,height".

Parameter	Type
G	Text / expression
x,y,width,height	Text / expression

Example

```
sgneoDataGridMove "[G]" "20,20,700,400"
```

12. sgneoDataGridDestroy

Destroy a grid.

Parameter	Type
G	Text / expression

Example

```
sgneoDataGridDestroy "[G]"
```

13. sgneoDataGridDestroyAll

Destroy every grid.

Parameter	Type
-----------	------

Example

```
sgneoDataGridDestroyAll
```

14. sgneoDataGridSetEditable

Turn in-place cell editing on (1) or off (0).

Parameter	Type
G	Text / expression
1/0	Text / expression

Example

```
sgneoDataGridSetEditable "[G]" "1"
```

15. sgneoDataGridSetKey

Set the table and primary-key column(s) that Save writes to. AutoInc (1/0) = key is server-generated. Required before saving edits.

Parameter	Type
G	Text / expression
Table	Text / expression
KeyColumn(s)	Text / expression

Parameter	Type
AutoInc(1/0)	Text / expression

Example

```
sgneoDataGridSetKey "[G]" "books" "id" "1"
```

16. sgneoDataGridSave

Write pending edits/inserts/deletes back to the database in one transaction; stores rows affected.

Parameter	Type
G	Text / expression
[Var]	Result variable

Example

```
sgneoDataGridSave "[G]" "[Written]"
```

Grid Save note: call sgneoDataGridSetKey before Save. For auto-increment keys, leave the generated key blank in new rows.

17. sgneoDataGridCancelChanges

Discard all pending edits and revert the grid.

Parameter	Type
G	Text / expression

Example

```
sgneoDataGridCancelChanges "[G]"
```

18. sgneoDataGridPendingChanges

Store the number of unsaved changed rows.

Parameter	Type
G	Text / expression
[Var]	Result variable

Example

```
sgneoDataGridPendingChanges "[G]" "[Changes]"
```

19. sgneoDataGridInsertRow

Add a new empty row for the user to fill in (grid must be editable).

Parameter	Type
G	Text / expression

Example

```
sgneoDataGridInsertRow "[G]"
```

20. sgneoDataGridDeleteRow

Delete the currently selected row (applied on Save).

Parameter	Type
G	Text / expression

Example

```
sgneoDataGridDeleteRow "[G]"
```

21. sgneoDataGridOnChange

When a row is edited: set a variable to the pending-change count and run a Subroutine.

Parameter	Type
G	Text / expression
[CountVar]	Result variable
Subroutine	Text / expression

Example

```
sgneoDataGridOnChange "[G]" "[Changes]" "OnGridChange"
```

22. sgneoDataGridSetColor

Set the grid background colour (name like clWhite or a hex value like \$00F0F0F0 / #RRGGBB).

Parameter	Type
G	Text / expression
Colour	Text / expression

Example

```
sgneoDataGridSetColor "[G]" "#FFFFFF"
```

23. sgneoDataGridSetFont

Set the data font: name, size, colour.

Parameter	Type
G	Text / expression
FontName	Text / expression
FontSize	Numeric value
TextColour	Text / expression

Example

```
sgneoDataGridSetFont "[G]" "Segoe UI" "10" "clBlack"
```

24. sgneoDataGridSetTitleColors

Set the header band colours: background, text.

Parameter	Type
G	Text / expression
HeaderBackground	Text / expression
HeaderText	Text / expression

Example

```
sgneoDataGridSetTitleColors "[G]" "#DDEBFF" "#123456"
```

25. sgneoDataGridSetTitleFont

Set the header font: name, size, bold (1/0).

Parameter	Type
G	Text / expression
FontName	Text / expression
FontSize	Numeric value
Bold	Text / expression

Example

```
sgneoDataGridSetTitleFont "[G]" "Segoe UI" "10" "1"
```

26. sgneoDataGridSetRowColors

Set row colour and alternate-row colour (same value on both = no striping).

Parameter	Type
G	Text / expression
RowColour	Text / expression
AlternateRowColour	Text / expression

Example

```
sgneoDataGridSetRowColors "[G]" "#FFFFFF" "#F6F0E8"
```

27. sgneoDataGridSetSelectionColors

Set the selected-row colours: background, text.

Parameter	Type
G	Text / expression
SelectionBackground	Text / expression
SelectionText	Text / expression

Example

```
sgneoDataGridSetSelectionColors "[G]" "#CCE5FF" "#000000"
```

28. sgneoDataGridSetGridLines

Show grid lines (1/0) and set their colour.

Parameter	Type
G	Text / expression
ShowLines	Text / expression
LineColour	Text / expression

Example

```
sgneoDataGridSetGridLines "[G]" "1" "clSilver"
```

29. sgneoDataGridSetRowHeight

Set the row height in pixels.

Parameter	Type
G	Text / expression

Parameter	Type
Height(px)	Numeric value

Example

```
sgneoDataGridSetRowHeight "[G]" "24"
```

5. neoChart — Interactive Charts

12 commands are registered in the latest source build for this component.

1. sgneoChartCreate

Create a chart inside an existing Rectangle object. Choose the Rectangle, then store the chart handle.

Parameter	Type
DB	Text / expression
RectangleName	VisualNEO object name
[Var]	Result variable

Example

```
sgneoChartCreate "[DB]" "ChartRect" "[C]"
```

2. sgneoChartLoad

Run a SELECT and plot it. Type = bar/hbar/stackedbar/line/area/point/pie. Params as in neoDatabaseQuery.

Parameter	Type
C	Text / expression
SQL	Text / expression
Params	Text / expression
Type	Text / expression

Example

```
sgneoChartLoad "[C]" "SELECT head,SUM(amount) FROM fee_payments GROUP BY head" "" "bar"
```

3. sgneoChartSetType

Change the chart type: bar, hbar, stackedbar, line, area, point, pie.

Parameter	Type
C	Text / expression
Type	Text / expression

Example

```
sgneoChartSetType "[C]" "line"
```

4. sgneoChartSetColumns

Set the label column and value column(s). Value columns semicolon-separated for multi-series.

Parameter	Type
C	Text / expression
LabelColumn	Text / expression
ValueColumns	Text / expression

Example

```
sgneoChartSetColumns "[C]" "head" "sem1;sem2;sem3"
```

5. sgneoChartSetTitle

Set the chart title (blank to hide).

Parameter	Type
C	Text / expression
Title	Text / expression

Example

```
sgneoChartSetTitle "[C]" "Fees by Head"
```

6. sgneoChartSetLegend

Show (1) or hide (0) the legend.

Parameter	Type
C	Text / expression
1/0	Text / expression

Example

```
sgneoChartSetLegend "[C]" "1"
```

7. sgneoChartRefresh

Re-run the chart's last query and redraw.

Parameter	Type
C	Text / expression

Example

```
sgneoChartRefresh "[C]"
```

8. sgneoChartSave

Save the chart to an image file (.jpg, .bmp or .wmf).

Parameter	Type
C	Text / expression
ImageFile	File name/path

Example

```
sgneoChartSave "[C]" "[PubDir]chart.jpg"
```

9. sgneoChartShow

Show the chart.

Parameter	Type
C	Text / expression

Example

```
sgneoChartShow "[C]"
```

10. sgneoChartHide

Hide the chart.

Parameter	Type
C	Text / expression

Example

```
sgneoChartHide "[C]"
```

11. sgneoChartDestroy

Destroy a chart.

Parameter	Type
C	Text / expression

Example

```
sgneoChartDestroy "[C]"
```

12. sgneoChartDestroyAll

Destroy every chart.

Parameter	Type
-----------	------

Example

```
sgneoChartDestroyAll
```

6. neoPDF — PDF Reports

3 commands are registered in the latest source build for this component.

1. sgneoPDFSetOption

Set all report options in one dialog (Title, Subtitle, Orientation, Page Size, Font, Font Size, Columns, Header Image) for this [DB]. Call before neoPDFReport.

Parameter	Type
DB	Text / expression
Title	Text / expression
Subtitle	Text / expression
Orientation	Text / expression
PageSize	Text / expression
Font	Text / expression
FontSize	Text / expression
Columns	Text / expression
HeaderImage	File name/path

Example

```
sgneoPDFSetOption "[DB]" "Fee Report" "Department of LIS" "landscape" "a4" "Segoe UI" "10"  
"head=Head;amount=Amount" "[PubDir]logo.png"
```

Call this before sgneoPDFReport. Report settings are associated with the database handle and reused by the report/print layer.

2. sgneoPDFReport

Run a SELECT and write a table report to a PDF file, using the options set with neoPDFSetOption for this [DB].

Parameter	Type
DB	Text / expression
SQL	Text / expression
Params	Text / expression
OutputPDF	File name/path

Example

```
sgneoPDFReport "[DB]" "SELECT * FROM fee_payments ORDER BY id" "" "[PubDir]report.pdf"
```

3. sgneoPDFResetOptions

Reset all report options for a [DB] handle back to defaults.

Parameter	Type
DB	Text / expression

Example

```
sgneoPDFResetOptions "[DB]"
```

7. neoPrint — Preview and Printing

3 commands are registered in the latest source build for this component.

1. sgneoPrintPreview

Show a print-preview window for a table report from a SELECT (uses neoPDFSetOption settings).

Parameter	Type
DB	Text / expression
SQL	Text / expression
Params	Text / expression

Example

```
sgneoPrintPreview "[DB]" "SELECT * FROM fee_payments" ""
```

2. sgneoPrintReport

Print a table report from a SELECT to the printer (shows the printer dialog).

Parameter	Type
DB	Text / expression
SQL	Text / expression
Params	Text / expression

Example

```
sgneoPrintReport "[DB]" "SELECT * FROM fee_payments" ""
```

3. sgneoPrintChart

Print a chart (created with neoChartCreate) to the printer.

Parameter	Type
C	Text / expression

Example

```
sgneoPrintChart "[C]"
```

8. neoMail — SMTP / IMAP / POP3

10 commands are registered in the latest source build for this component.

1. sgneoMailConfigure

Set the mail account: host, port, username, password, security (none/starttls/ssl), from-address.

Parameter	Type
Host	Text / expression
Port	Numeric value
User	Text / expression
Password	Text / expression
Security	Text / expression
From	Text / expression

Example

```
sgneoMailConfigure "smtp.gmail.com" "587" "user@example.com" "APP_PASSWORD" "starttls" "user@example.com"
```

For SMTP TLS, use the security value documented by the plug-in such as `starttls` or `ssl`, with the appropriate server port.

2. sgneoMailSend

Send a simple e-mail (to, subject, body) using the configured account.

Parameter	Type
To	Text / expression
Subject	Text / expression
Body	Text / expression

Example

```
sgneoMailSend "recipient@example.com" "Subject" "Body text"
```

3. sgneoMailSendEx

Send an e-mail with cc, bcc, attachments (semicolon-separated paths) and html flag (1/0).

Parameter	Type
To	Text / expression
Cc	Text / expression
Bcc	Text / expression
Subject	Text / expression
Body	Text / expression
Attachments(;)	Text / expression
HTML(1/0)	Text / expression

Example

```
sgneoMailSendEx "to@example.com" "cc@example.com" "bcc@example.com" "Subject" "Body text" "[PubDir]report.pdf" "0"
```

4. sgneoMailAttach

Queue a file to attach to the next send.

Parameter	Type
File	File name/path

Example

```
sgneoMailAttach "[PubDir]report.pdf"
```

5. sgneoMailClearAttachments

Clear all queued attachments.

Parameter	Type
-----------	------

Example

```
sgneoMailClearAttachments
```

6. sgneoMailIMAPCount

Store the number of messages in an IMAP mailbox (blank = INBOX).

Parameter	Type
Mailbox	Text / expression
[Var]	Result variable

Example

```
sgneoMailIMAPCount "INBOX" "[Count]"
```

7. sgneoMailIMAPHeader

Store an IMAP message's headers (From/Subject/Date).

Parameter	Type
Mailbox	Text / expression
MsgNo	Numeric value
[Var]	Result variable

Example

```
sgneoMailIMAPHeader "INBOX" "1" "[Header]"
```

8. sgneoMailIMAPBody

Store an IMAP message's body.

Parameter	Type
Mailbox	Text / expression
MsgNo	Numeric value
[Var]	Result variable

Example

```
sgneoMailIMAPBody "INBOX" "1" "[Body]"
```

9. sgneoMailPOP3Count

Store the number of messages waiting on the POP3 server.

Parameter	Type
[Var]	Result variable

Example

```
sgneoMailPOP3Count "[Count]"
```

10. sgneoMailPOP3Message

Store a full POP3 message (headers + body).

Parameter	Type
MsgNo	Numeric value
[Var]	Result variable

Example

```
sgneoMailPOP3Message "1" "[Message]"
```

9. neoCrypto — Hashing, Encoding, AES and Passwords

18 commands are registered in the latest source build for this component.

1. sgneoCryptoMD5

Store the MD5 hash of a string.

Parameter	Type
Text	Text / expression
[Var]	Result variable

Example

```
sgneoCryptoMD5 "text" "[H]"
```

2. sgneoCryptoSHA1

Store the SHA-1 hash of a string.

Parameter	Type
Text	Text / expression
[Var]	Result variable

Example

```
sgneoCryptoSHA1 "text" "[H]"
```

3. sgneoCryptoSHA256

Store the SHA-256 hash of a string.

Parameter	Type
Text	Text / expression
[Var]	Result variable

Example

```
sgneoCryptoSHA256 "text" "[H]"
```

4. sgneoCryptoSHA512

Store the SHA-512 hash of a string.

Parameter	Type
Text	Text / expression
[Var]	Result variable

Example

```
sgneoCryptoSHA512 "text" "[H]"
```

5. sgneoCryptoFileSHA256

Store the SHA-256 hash of a file (integrity check).

Parameter	Type
File	File name/path
[Var]	Result variable

Example

```
sgneoCryptoFileSHA256 "[PubDir]report.pdf" "[H]"
```

6. sgneoCryptoHMACSHA256

Store the HMAC-SHA256 of a message with a key.

Parameter	Type
Key	Text / expression
Message	Text / expression
[Var]	Result variable

Example

```
sgneoCryptoHMACSHA256 "secret" "message" "[H]"
```

7. sgneoCryptoHMACSHA1

Store the HMAC-SHA1 of a message with a key.

Parameter	Type
Key	Text / expression
Message	Text / expression
[Var]	Result variable

Example

```
sgneoCryptoHMACSHA1 "secret" "message" "[H]"
```

8. sgneoCryptoBase64Encode

Base64-encode a string.

Parameter	Type
Text	Text / expression
[Var]	Result variable

Example

```
sgneoCryptoBase64Encode "Hello" "[B64]"
```

9. sgneoCryptoBase64Decode

Base64-decode a string.

Parameter	Type
Text	Text / expression
[Var]	Result variable

Example

```
sgneoCryptoBase64Decode "[B64]" "[Plain]"
```

10. sgneoCryptoHexEncode

Hex-encode a string.

Parameter	Type
Text	Text / expression
[Var]	Result variable

Example

```
sgneoCryptoHexEncode "Hello" "[Hex]"
```

11. sgneoCryptoHexDecode

Hex-decode a string.

Parameter	Type
Text	Text / expression
[Var]	Result variable

Example

```
sgneoCryptoHexDecode "[Hex]" "[Plain]"
```

12. sgneoCryptoAESEncrypt

AES-256 encrypt a string with a password; stores base64 result.

Parameter	Type
Text	Text / expression
Password	Text / expression
[Var]	Result variable

Example

```
sgneoCryptoAESEncrypt "sensitive data" "password" "[Cipher]"
```

13. sgneoCryptoAESDecrypt

AES-256 decrypt a base64 string with a password.

Parameter	Type
Text	Text / expression
Password	Text / expression
[Var]	Result variable

Example

```
sgneoCryptoAESDecrypt "[Cipher]" "password" "[Plain]"
```

14. sgneoCryptoAESEncryptFile

AES-256 encrypt a file with a password.

Parameter	Type
InFile	File name/path
OutFile	File name/path
Password	Text / expression

Example

```
sgneoCryptoAESEncryptFile "[PubDir]input.dat" "[PubDir]input.enc" "password"
```

15. sgneoCryptoAESDecryptFile

AES-256 decrypt a file with a password.

Parameter	Type
InFile	File name/path
OutFile	File name/path
Password	Text / expression

Example

```
sgneoCryptoAESDecryptFile "[PubDir]input.enc" "[PubDir]output.dat" "password"
```

16. sgneoCryptoPasswordHash

Hash a password (PBKDF2-SHA256, salted) for safe storage.

Parameter	Type
Password	Text / expression
[Var]	Result variable

Example

```
sgneoCryptoPasswordHash "secret" "[Hash]"
```

17. sgneoCryptoPasswordVerify

Verify a password against a stored hash; stores True/False.

Parameter	Type
Password	Text / expression
StoredHash	Text / expression
[Var]	Result variable

Example

```
sgneoCryptoPasswordVerify "secret" "[Hash]" "[OK]"
```

18. sgneoCryptoRandomToken

Generate a random hex token of N bytes (e.g. session id, salt).

Parameter	Type
Bytes	Numeric value
[Var]	Result variable

Example

```
sgneoCryptoRandomToken "16" "[Token]"
```

10. Form Builder — Generated Data Entry Forms

3 commands are registered in the latest source build for this component.

1. sgneoFormBuilder

Auto-create a data-entry form from a table. Pick a Rectangle to hold the form, then select a table.

Parameter	Type
DB	Text / expression
RectangleName	VisualNEO object name
[Var]	Result variable

Example

```
sgneoFormBuilder "[DB]" "FormRect" "[Form]"
```

The form builder creates a data-entry form in the selected Rectangle and returns a form handle. Use the navigation/edit actions against the same database/table session.

2. sgneoFormDestroy

Destroy a form created by sgneoFormBuilder.

Parameter	Type
FormHandle	Text / expression

Example

```
sgneoFormDestroy "[Form]"
```

3. sgneoFormDestroyAll

Destroy all forms created by sgneoFormBuilder.

Parameter	Type
-----------	------

Example

```
sgneoFormDestroyAll
```

Practical Recipes

Transactional update

```
sgneoDatabaseBeginTransaction "[DB]"
sgneoDatabaseExecute "[DB]" "UPDATE books SET price=? WHERE id=?" "[Price|BookId]" "[Affected]"
sgneoDatabaseCommit "[DB]"
```

Prepared statement workflow

```
sgneoDatabasePrepare "[DB]" "UPDATE books SET price=? WHERE id=?" "[Stmt]"
sgneoDatabaseExecutePrepared "[Stmt]" "[Price|BookId]" "[Affected]"
sgneoDatabaseFinalize "[Stmt]"
```

Editable grid workflow

```
sgneoDataGridCreate "[DB]" "Grid1" "[G]"
sgneoDataGridLoad "[G]" "SELECT * FROM books" ""
sgneoDataGridSetEditable "[G]" "1"
sgneoDataGridSetKey "[G]" "books" "id" "1"
sgneoDataGridSave "[G]" "[Written]"
```

Chart + PDF

```
sgneoChartCreate "[DB]" "ChartRect" "[C]"
sgneoChartLoad "[C]" "SELECT head,SUM(amount) FROM fee_payments GROUP BY head" "" "bar"
sgneoChartSave "[C]" "[PubDir]chart.jpg"
sgneoPDFReport "[DB]" "SELECT * FROM fee_payments ORDER BY id" "" "[PubDir]report.pdf"
```

Error handling

```
sgneoDatabaseQuery "[DB]" "SELECT * FROM books" "" "[R]"
If "[neoDatabaseError]" "<>" ""
AlertBox "Error" "[neoDatabaseError]"
Return
EndIf
```

JSON result extraction

```
sgneoDatabaseQuery "[DB]" "SELECT id,title FROM books" "" "[R]"
sgneoDatabaseJSONGet "[R]" "0.title" "[First]"
```

Mail with attachment

```
sgneoMailAttach "[PubDir]report.pdf"
sgneoMailSend "recipient@example.com" "Report" "Please find the report attached."
```

Troubleshooting and Compatibility

Database connection failures

For PostgreSQL and MySQL/MariaDB, verify that the client DLL is Win32/x86 and that every dependent DLL is available. A 64-bit client library cannot be loaded by this Win32 plug-in.

PostgreSQL schema naming

PostgreSQL tables normally reside in the public schema. Use a table name such as applications or qualify it as public.applications when SQL requires explicit qualification. Do not treat the database name itself as the schema.

Unicode

The latest development line includes UTF-16 boundary sanitization intended to prevent VisualNEO errors caused by unpaired UTF-16 surrogate characters in database results or metadata.

Grid Save failures

Ensure the grid is editable, call SetKey with the correct table/key, and use AutoInc=1 when the key is server-generated. Check the pending-change count before Save when diagnosing an edit workflow.

Mail TLS

Use the 32-bit OpenSSL 3 pair supplied for neoMail TLS. For Gmail, use an App Password rather than a normal account password when required by the account configuration.

Errors

The source documents use [neoDatabaseError] and [LastError] as error reporting variables. Applications should check these after operations that can fail.

Command Coverage Summary

Component	Registered actions
neoDatabase	49
neoDataGrid	29
neoChart	12
neoPDF	3
neoPrint	3
neoMail	10
neoCrypto	18
Form Builder	3
TOTAL	127

Source basis

This help file was generated from the latest supplied sgNeoDatabase.nbp, the corresponding latest source build, and the previously prepared command documentation. Command names, parameter counts and descriptions are grounded in the current source registrations; examples are explicitly illustrative where the source does not provide a complete application-specific workflow.

End of continuous help file.