



# 4D v11 SQL Release 4 (11.4)

## ADDENDUM

---

Welcome to release 4 of 4D v11 SQL. This document presents the new features and the modifications made to this new version of the program.

### Language

#### New Parameter for Menu Management Commands

The syntax of three "Menus" theme commands has been modified: **APPEND MENU ITEM**, **INSERT MENU ITEM** and **SET MENU ITEM**. These commands now accept an asterisk (\*) as the last parameter.

**APPEND MENU ITEM** (menu; itemText{; subMenu{; process}}{;.\*})

**INSERT MENU ITEM** (menu; afterItem; itemText{; subMenu{; process}}{;.\*})

**SET MENU ITEM** (menu; menuItem; itemText{; process}}{;.\*})

Parameter	Type	Description
*	→	If passed: consider metacharacters as simple strings

In these three commands, the \* parameter indicates, when it is passed, that "special" characters included in the menu item label must be considered as standard characters and not as *metacharacters*.

These characters are, for example, "(", "!" or ";" (which can be used, respectively, to disable the item, to add a checkbox or to create several items). For a complete list of metacharacters, please refer to the description of the **APPEND MENU ITEM** command.

If you pass the \* parameter, these characters will be handled as standard characters, which means that you can create items with labels such as **"Copy (special)..."** or **"Find/Replace..."**.

Note that when the \* parameter is passed, you cannot create several items in a single call since the ";" character will be considered as a standard character.

---

*Note:* The GET MENU ITEMS and Get menu item commands will return or not return the metacharacters of a label according to how it was created: if it was created with the \* option, the metacharacters will be returned as standard characters.

---

## Renamed Commands

For better clarity, three commands of the "Menus" theme have been renamed in 4D v11 SQL release 4: SET MENU ITEM REFERENCE, Get menu item reference and Get selected menu item reference. In fact, the wording of these command names can lead to confusion because they do not work with *internal* menu references (unique identifiers of the *MenuRef* type) but instead work with *user* references (custom character strings), mainly used by the Dynamic pop up menu command. Remember that user references can be set in the Menu editor ("Reference" field).

These commands have been renamed as follows:

Former name	New name in v11.4
<i>SET MENU ITEM REFERENCE</i>	SET MENU ITEM PARAMETER
<i>Get menu item reference</i>	Get menu item parameter
<i>Get selected menu item reference</i>	Get selected menu item parameter

As a consequence, the documentation of these commands has been modified. Their syntax remains unchanged. They will be renamed automatically in the methods of existing databases.

## SET DATABASE PARAMETER, Get database parameter

A new selector (constant) is available for the SET DATABASE PARAMETER and Get database parameter commands.

**Selector = 54** (Idle Connections Timeout)

- *Values:* Whole value expressing a duration in seconds. The value can be positive (new connections) or negative (existing connections). By default, the value is 0 (no timeout).

- *Description:* This parameter can be used to set the maximum period of inactivity (timeout) for connections to both the 4D database engine and the SQL engine. It applies to all the connections opened by the machine where the command was executed.

When an idle connection reaches this limit, it is automatically put on standby, which freezes the client/server session and closes the network socket. This functioning is completely transparent for the user: as soon as there is new activity on the connection that is on standby, the socket is automatically reopened and the client/server session is restored.

On the one hand, this setting lets you save resources on the server: connections on standby close the socket and free up a process on the server. On the other hand, it also lets you avoid losing connections due to closing of idle sockets by the firewall. For this, the timeout value for idle connections must be lower than that of the firewall in this case.

If you pass a positive value in *value*, it will apply to all new connections in all the processes. If you pass a negative value, it will apply to connections that are open in the current process. If you pass 0, idle connections will not be subjected to a timeout.

## SQL SET OPTION

SQL SET OPTION(option; value)

The SQL SET OPTION command now accepts "WCHAR" as a *value* for the SQL Charset option. This particular value must be passed as a string:

**SQL SET OPTION(SQL Charset;"WCHAR")**

When this statement is executed, the encoding used by the 4D SQL server is automatically adapted to the running platform:

- Under Windows, UTF-16 is used,
- Under Mac OS, UTF-32 is used.

## Renaming of "System Folder" Constants

Several constants of the "System Folder" theme have been renamed or prefixed for the following reasons:

- Evolution of OS system folder names (*Preferences* or *Profiles* type),
- Constants that no longer work under Mac OS (*Startup Items*, *Apple* or *Start Menu* type),
- Constants suffixed "Win" but which now function under Mac OS as well (*Desktop Win* and *Program Files Win*),

- System folders that are no longer used (constants prefixed \_O\_).

Former name	Value	New name in v11.4
Preferences or Profiles_All	2	User Preferences_All
Preferences or Profiles_User	3	User Preferences_User
Startup Items_All	4	Startup Win_All
Startup Items_User	5	Startup Win_User
Apple or Start Menu_All	8	Start Menu Win_All
Apple or Start Menu_User	9	Start Menu Win_User
Desktop Win	15	Desktop
Program Files Win	16	Applications or Program Files
Mac Shutdown Items_All	6	_O_Mac Shutdown Items_All
Mac Shutdown Items_User	7	_O_Mac Shutdown Items_User
Mac Extensions	10	_O_Mac Extensions
Mac Control Panels	11	_O_Mac Control Panels

## Design Mode

### Disabled Check Boxes in List Boxes







In listboxes, it is possible to display *disabled* (grayed out) three-states check boxes.

Remember that in order to display check boxes in a list box, you must choose **Three-states checkbox** for the "Display Type" column property. The number values (0, 1 or 2) stored in the column will then be shown as active check boxes.

In order to display disabled check boxes (which cannot be modified by the user), you can simply pass negative values:

- -4 = disabled filled or semi-checked box (intermediate state)
- -3 = disabled checked box
- -2 = disabled unchecked box
- -1 = invisible check box

The following picture illustrates the different states:

Appearance	Value	Type
	-4	Disabled filled (semi-checked)
	-3	Disabled checked
	-2	Disabled unchecked
	-1	Invisible
	0	Unchecked
	1	Checked
	2	Filled (semi-checked)

## Erratum

Contrary to what is indicated in the *Design Reference* manual, the "Internal storage max size" option of the Inspector does not apply to BLOB and Picture type fields. These types of data are always stored outside of records. In version 11.4 of 4D, this option no longer appears for BLOB and Picture fields.

## Support of DEP (Windows)

4D v11 SQL release 4 applications are compliant with DEP (Data Execution Prevention) specifications. Designed by Microsoft and included in Windows beginning with the Windows XP SP 2, Windows Vista, and Windows Server 2003 versions, DEP "is a set of hardware and software technologies that perform additional checks on memory to help prevent malicious code from running on a system". The principle consists in preventing code execution from data pages. For more information, please refer to: <http://support.microsoft.com/kb/875352>.

To be able to work in systems where DEP is enabled, applications must be compliant with this system or be included in the list of applications that are exempted from this control (when this setting is authorized). Release 4 of 4D v11 SQL lets you execute your 4D applications under the control of DEP. No additional configuration is required. Note that any plug-ins used must also be compliant with DEP.

## 4D Pack

A new command is available in 4D Pack ([AP Modify method](#)) and the existing [AP Create method](#) command has been updated. These two commands are found in the "4D Pack: Utilities" theme.

### AP Modify method

AP Modify method (methodName; methodCode) → Longint

Parameter	Type		Description
methodName	String (31)	→	Name of method to modify
methodCode	BLOB	→	BLOB containing the method text
Function result	Longint	←	Error code (0=no error)

The new AP Modify method command lets you modify the source code of a project method in the structure of a 4D database (interpreted databases only).

Pass the name of an existing project method in the *methodName* parameter (the names of 4D methods may contain up to 31 characters).

In *methodCode*, pass a BLOB containing the new text of the method. If you use the TEXT TO BLOB command to fill the BLOB, specify the Mac C string format (or 0) as the 3rd parameter of this command.

If the method is modified correctly, the command returns 0.

Otherwise, it returns one of the following error codes:

1 = A bad parameter type has been passed.

2 = The database is running in compiled mode.

3 = The method name is an empty string.

4013 = Invalid method name.

4015 = The method source code could not be modified.

---

*Note :* This command does not work if it is executed from a compiled application.

---

## AP Create method

AP Create method (*methodName*; *propertiesArray*; *methodCode*{; folder-Name}) → Longint

The *propertiesArray* array must contain 7 rows: pass 1 in *propertiesArray*{7} if the method must have the "Execute on Server" attribute, and 0 otherwise.

In *methodCode*, you can pass the text of a method that has been stored in a file via the **Export Method...** menu command (in this case, the text includes commands, constants, etc., as references (*tokens*) and not as words).

Also note that this command can return the following error codes:

4013 = Invalid method name.

4014 = Could not create method.

## 4D SVG Component

Three commands have been updated and the *strokeWidth* parameter of the drawing commands now accepts Real values.

**SVG\_SET\_DIMENSIONS** SVG\_SET\_DIMENSIONS (svgObject; width{; height{; unit}})

The *unit* parameter of the SVG\_SET\_DIMENSIONS command now accepts the following additional values: em, ex or %.

**SVG\_SET\_STROKE\_WIDTH** SVG\_SET\_STROKE\_WIDTH (svgObject; strokeWidth{; unit})

Parameter	Type		Description
svgObject	SVG_Ref	→	Reference of SVG element
strokeWidth	Real	→	Line thickness
unit	String	→	Unit of measurement

A new optional *unit* parameter can be passed which accepts one of the following values: em, ex, px, pt, pc, cm, mm, in or %. An error is generated if you pass an incorrect value.

**SVG\_PATH\_ARC** SVG\_PATH\_ARC (parentSVGObject; xRadius; yRadius; x; y{; rotation{; arc-path}})

Parameter	Type		Description
parentSVGObject	SVG_Ref	→	Reference of path element
xRadius	Number	→	Radius of ellipse on X axis
yRadius	Number	→	Radius of ellipse on Y axis
x	Number	→	Coordinate on X axis of destination point
y	Number	→	Coordinate on Y axis of destination point
rotation	Number	→	Value of rotation
arcpath	Longint	→	Sets the way the arc will be drawn

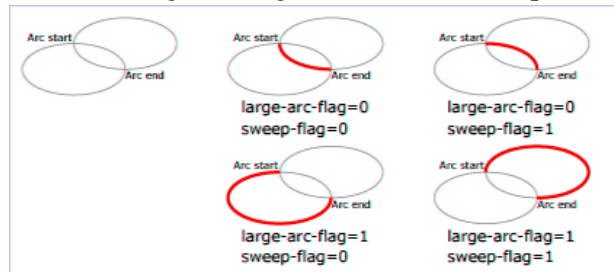
The new optional *arcpath* parameter can be used to apply a combination of constraints which will determine how the arc will be drawn. The large-arc-flag constraint is used to choose (or not) the larger of the two possible arcs (greater than 180°) and the sweep-flag constraint chooses the direction it will be drawn (positive angle or negative angle).

The following values, representing the four possible combinations of the two constraints, can be passed:

- 0: large-arc-flag = 0, sweep-flag = 1
- 1: large-arc-flag = 1, sweep-flag = 0
- 2: large-arc-flag = 0, sweep-flag = 0
- 3: large-arc-flag = 1, sweep-flag = 1

When large-arc-flag is equal to 1, the larger arc is drawn (and the smaller when it is equal to 0). When sweep-flag is equal to 1, the arc is drawn at a positive angle (and at a negative angle when it is equal to 0).

The following drawing illustrates the four possible combinations:



By default, the value of *arcpath* is 0 (large-arc-flag=0, sweep-flag=1).

**Drawing Commands** The *strokeWidth* parameter now accepts a Real value instead of a Longint. This permits numbers less than 1, such as 0.5, to be passed. This modification concerns the following commands:

SVG\_New\_arc  
 SVG\_New\_circle  
 SVG\_New\_ellipse  
 SVG\_New\_ellipse\_bounded  
 SVG\_New\_line  
 SVG\_New\_path  
 SVG\_New\_polygon  
 SVG\_New\_polygon\_by\_arrays  
 SVG\_New\_polyline  
 SVG\_New\_polyline\_by\_arrays  
 SVG\_New\_rect  
 SVG\_New\_regular\_polygon  
 SVG\_SET\_STROKE\_WIDTH