DEPRECATED FEATURES

4D v12 and higher

Rev. 03 Jun 2010

Table of contents

Introduction

For over 25 years, our main goal has been to enhance the product (using new concepts and technologies) while ensuring compatibility of 4D applications. As early adopters of 4D know, we always have put a lot of effort into compatibility, and we can find around the world 4D applications that were created years and years ago, with old versions of 4D and of an OS, that are still working with the latest revisions of the product.

Unfortunately, it sometimes becomes too difficult to mix old technologies with new ones:

- 4D *must* bring new technologies, new APIs, new paradigms to the developers
- OSs change every day, and sometime deprecate their own old APIs.

This is why 4D sometimes needs to tag some commands and features as deprecated: which means that, one day, the command will be removed from the language in a future major version. Letting the developer know what is deprecated and what kind of replacement can be used makes it much more comfortable for him/her to apply the change in the code: there is no urgency, no pressure, the developer has plenty of time to change the code.

We'll start with a quick list of the deprecated functionalities, followed by details about each of them. Then, a list of deprecated commands is established.

Quick table

Possible values for the “Status” column:

Next	we plan to remove the feature in the next major version after v12
Later	will be removed in a future major version
OS	depends on officially deprecated OS technologies (e.g.: QuickDraw, PICT format). Status is the same as *Later*, but an OS could remove the support before us.

Theme	Replace with	Status
4D Chart (plug-in area)	SVG / webarea / PHP	*Later*
4D Draw	SVG	*(Removed since v11)*
4D Open	Synchronization, WebServices, SQL Pass Thru	*(Removed since v11)*
PICT	Modern formats	*OS*
QuickDraw Patterns (form object property)	SVG / picture using the “Replicated” format	*Next*
Mac Resources	Use the “Resources” folder	*OS* We plan to remove “write” access in a first step (*Next*)
Old Looks	Use System looks	*Next*
Altura Mac2Win	Plug-in developers: use native Window code	*Next*
SubTables	Use N->1 tables	*Later*
Web Contextual Mode	Use non-contextual mode	*Next*
Non Unicode mode	Move to Unicode	*Later*

4D Chart (using the plug-in area)

Note
getting a graph in an SVG picture (new available syntax since 4D v11) is the recommended way to create a graph. The deprecation concerns the usage of the 4D Chart Plug-in Area and its commands.

Since version 11 of 4D, it is possible to call the `GRAPH()` command, passing it a variable of the Picture type, instead of a 4D Chart area reference. So, developers who just used the `GRAPH` command to display or print a graph can easily move to SVG: just use a picture variable instead of a 4D Chart plug-in area reference.

There are a lot of advantages to using SVG:

- It's *very powerful*: gradients, transparency, layouts, rotations, shadows, …
- It's vectorial: can be resized with no "pixelization"
- It uses modern drawing APIs of the operating system
- It is text (basically, xml): it can be easily manipulated
- All "Images" theme commands work with SVG
- An SVG picture can be displayed in a WebArea (IE joined the SVG group in 2010, a plug-in will no longer be needed in IE 9)
- Native 4D commands and *4D SVG* component methods can be used to let the end user interact with the SVG.

All these properties will let the developers build professional and nice-looking graphs.
If the 4D Chart Plug-in Area was used to let the user manipulate the graph (change colors, change lines, add comments, …), then – again – SVG is a nice replacement solution: using the tools provided with 4D, the developer will be able to create dynamic graphs in SVG.

Besides SVG, there are other ways to generate graphs: WebArea + JavaScript, the new PHP Execute command that opens the door to a lot of public code (for example, refer to the "Flash Your Charts" 4D Summit 2009 conference). The Web 2.0 Pack also comes with commands to draw graphs in a browser (using SVG too).

4D Draw

Note
4D Draw is deprecated since 4D version 11, and is already no longer available in 4D version 12.

4D Draw should be replaced with SVG. As said in the previous chapter about 4D Chart, using SVG has a lot of advantages (powerful effects, modern, text, etc.). The 4D language and the 4D SVG component v12 can be used for this purpose. In the demo databases (downloadable from our web site), you will find some example of user interaction within SVG (the SVG Theater demo for example).

Creating dynamic drawing content can be done with these tools. An alternative solution that can be analyzed is the use of JavaScript code (for example, the Raphael framework).

4D Open

Note
4D Open is unavailable since 4D version 11

4D Open was a plug-in which could connect and talk directly with 4D at the database level. Since 4D v11 SQL and the rewriting of the database engine, it does not work anymore.

4D Open was used for 2 main purposes:

- Multisite synchronization/replication
- Administration-maintenance (launch execution of maintenance methods)

Whatever the kind of use, v12 provides three main alternatives:

- The new replication/synchronization features available in SQL
- Enhanced WebServices
- SQL Pass Thru (since version 11.5)

PICT

The 'PICT' format is a very old Mac format. Prior to v11, 4D stored all pictures in this format, even on Windows. PICT is deprecated since QuickDraw has been deprecated (2005).

There is one important thing to understand about PICT. It can store (read "encapsulate") 2 main kinds of information:

- The drawing primitives themselves (either bitmap or vectorial)
- Or a more modern format (jpeg for example) stored in a PICT using QuickTime. Typically: the developer called `COMPRESS PICTURE` with the `QT Photo compressor` constant.

This means that even if all pictures stored in the data files before 4D v11 were PICT, those PICT could, in fact, contain JPEG. It's important for our customers to stop using PICT, mostly because 4D needs Altura (+ QuickTime if **`COMPRESS PICTURE`** were used) to read PICT on Windows: this is not efficient, and it requires QuickTime to be installed.

When migrating data from pre-v11 to v11, developers should apply the **`CONVERT PICTURE`** command to every picture field of the data. If the picture field, in fact, encapsulates a 4D Write/4D View area, he then uses **`CONVERT PICTURE`** with a specific codec (see the documentation of the command).

QuickDraw patterns

QuickDraw patterns are sometimes used in the form object properties palette window. Again, a replacement solution is to use SVG to create the patterns. Using SVG, the "replicated" display format, and the transparency, developers should be able to replace the QuickDraw patterns.

Mac Resources

Note
The v11 documentation contains a chapter about deprecation and compatibility of Mac resources

This is another old Mac OS technology, deprecated since Mac OS X.4 (Tiger, 2005). Resources are used to store structured data such as text and strings (localization), but also icons, etc. Basically, we can say that it's not the resources that are deprecated, it's their support on disk, the *Resource Fork*. The resource fork is part of the Mac OS file system, and since Mac OS X exists, Apple tries to remove this support since it is not compatible with other file systems (Unix, Windows), and is the source of a lot of problems when transferred via the network.

On Windows, this mechanism is emulated and Mac Resources reside in a .RSR file.

But even if there are still APIs to handle resources (and Mac OS transparently handles resources stored in a *data fork*), it is no longer recommended to use this old mechanism for several reasons:

- Text and strings are Mac-Roman. You can't store Unicode in resources of type TEXT or STR#
- PICT resources store PICTs: not modern, deprecated, no transparency, etc See chapter VI.
- The count of resources and the size of the resources are limited (about 2 700 resources or 16 MB)

What we plan to do is to first remove the support for commands that *write/create* resources.

The vast majority of 4D applications using resources are in fact using the "Strings List" resources, 'STR#'. 4D provides tools to easily move from STR# to XLIFF:

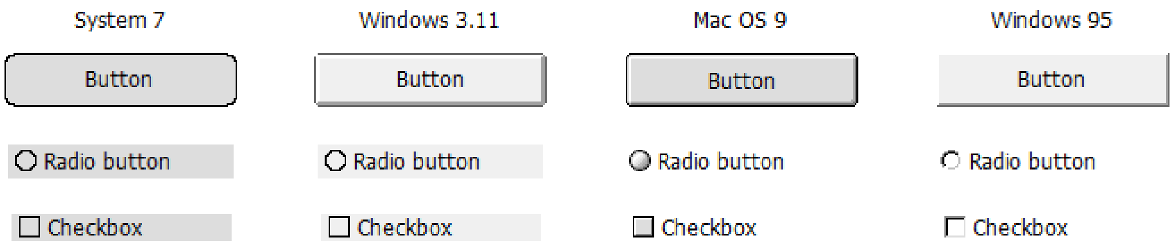
- The *4D Pop* Component can automatically create the XLIFF files by reading and transferring the content of the STR#.
- All the routines and expressions that reference STR# work with no change with XLIFF. For example, if the label of a button or a menu was ":15000,3" (meaning "get the third item of STR# ID 15000"), 4D will load the appropriate XLIFF (if it exists).

For other kinds of resources:

- Put resources in separate files inside the Resources folder (create sub-directories if needed):
  - Save 'TEXT' resources in XLIFF or .txt files
  - Save 'PICT' resources as separate .jpg/.png/etc. files
  - Save 'PICT' + MASK' resources as png files
  - Use (on Mac) icns instead of ICON or colored icons
  - Save any private resources as appropriate for you (typically: save as a binary file with a specific extension)
- Use the new (since v11) "Resources" folder to store your resources. Use **`Get 4D folder(`**`Current resources folder`**`)`** to dynamically get the parent path for your resources.

Old looks

What we call “Old Looks” are System 7, Windows 3.11, ….



Since several versions, 4D uses native controls (controls drawn by the OS). To handle old looks or cross-platform looks is no longer possible. We also think that Windows 7 users don’t want to see a Mac OS 7 button, and Mac users are allergic to Windows 3.11 objects: more and more end users want native interfaces, and each major version of 4D since 4D 2004 provides more and more *native* controls (for example: indeterminate progress bars, steppers in v12). And end users deserve modern interfaces. That is why:

- Since 4D 2004, these looks have been deprecated. New structures can’t use those old looks. The look for objects is “System” or “Printer”.
- Converted structures can still have those looks, but objects are then drawn using QuickDraw (instead of using native OS controls), handling those objects can’t use native APIs, etc.: we plan to remove this support for the next major version (> v12).

4D Developers will have to modify their forms, and to apply the “System” (or “Printer”) look to the appropriate objects (most of the time, the form itself).

Altura Mac2Win

Altura Mac2Win was used to port 4D to Windows. It is a set of APIs that helped porting a Mac OS (pre OS X) code to Windows, by translating APIs: filesystem, QuickDraw, Resources, PICT, etc. It was very useful and helped a lot (Mac plug-in developers, for example, could move their plug-ins to Windows more easily), but it translates old (read “deprecated”) Mac OS APIs, and doesn’t use modern native Windows APIs: 4D must remove Mac2Win from its code as much as possible. This is very long and hard work, and in each version of 4D, some dependencies are removed (and replaced by modern APIs).
Right now, 4D still depends on it in part, mostly to be able to handle compatibility of old databases: resources, PICT,   part of the user events handling, support for third party plug-ins that are built using Altura, etc.

By removing resources in the .RSR file to separate files in the “Resources” folder, and by converting (**`CONVERT PICTURE`**) to not-PICT, the 4D developer will be ready once 4D has removed Altura.
But the first people concerned by this huge step are plug-in developers: they must stop using Altura as soon as possible, which means they must rewrite some parts of their Windows source code (we have already warned them for several years).

Subtables

Since several major versions, 4D has warned developers against the use of subtables. Since v11, it is impossible to create a field of the SubTable type. Subrecords have some known limitations. For example, they are always loaded in memory; they are not handled by the **SEND RECORD** or **DUPLICATE RECORD** commands.

We don't plan to remove support for subtables in the near future, but it's really time for developers to convert their subtables to regular N-> tables because we plan to remove it in a future major version of 4D. Developers who used subtables for performance reasons (some specific situations in which loading related records was slow) can be reassured, especially with v12: the speed is here and using classic N<->1 relations is very fast.

Basically, there are two main ways to remove subtables (note: the following is not a full techtips; just a quick overview):

- *Before* conversion from a pre-v11 structure: in 2004, create the appropriate N table and the ID field in the 1 table (if not already there). Then change the code everywhere it is needed (see below).
- *After* conversion: in this situation, 4D has replaced the subtable with a N table using a special relation, that lets the language to work with the subselection and the subrecords. The 4D developer needs to remove this special relation, replace it with a normal relation and change the code everywhere it is needed (see below).

What we mean by "change the code everywhere it is needed" is, basically:

- Create the new forms, update included forms
- In the methods (project, form, object, etc.):
    - Replace all commands of the "SubRecords" theme with the corresponding Selection or Record command (for example, replace **CREATE SUBRECORD** with **CREATE RECORD**, filling the ID fields)
    - Explicitly load the N records when needed

Web Contextual mode

This mode is not compatible with a modern web server (for example, it is not possible to use the Back/Forward buttons of a browser, the interface is poor, etc.). Since we plan to integrate a brand new Web server in the next major version after v12, and this new Web Server will not be compatible with the Contextual Mode, we strongly encourage 4D developers using the web contextual mode to switch to the non-contextual mode, which means the code must be modified (including the code of HTML pages).

Non-unicode mode

Supporting ASCII mode (synonym for "non-Unicode mode") leads to poor performance when manipulating text because it must be converted to and from Mac Roman every time it is used in the legacy-converted-structure. We plan to remove ASCII mode in future major versions. Note that support for ASCII mode is already removed for compiled structures running under 4D Server 64-bits Windows.
4D developers should – for converted structures – activate the Unicode mode. The *Conversion to 4D v11 SQL.pdf* document (located in the v11 section of the documentation, at http://doc.4d.com) gives hints about this topic.

Language: deprecated commands

As a logical consequence of the deprecated technologies seen in the previous chapters, here is a list of deprecated commands. The list also contained other commands, such as **ADD DATA SEGMENT**, that are deprecated since previous versions of 4D.
4D developers are strongly encouraged to replace these commands with the new, supported feature. The detailed documentation of each command (http://doc.4d.com), when appropriate, gives hint about the replacement solution(s):

Theme	**Command**	**Replacement**
4D Environment	ADD DATA SEGMENT DATA SEGMENT LIST	(only one segment since 4D v11)
	Get 4D Folder and Extras Folder constant	Move your documents from the Extras folder to the Resources folder, change the code accordingly
Compiler	C_GRAPH	(use SVG with the GRAPH command)
	C_INTEGER	C_LONGINT
	C_STRING	C_TEXT as soon as the database is Unicode
	ARRAY STRING	ARRAY TEXT
Data entry	ADD SUBRECORD	ADD RECORD in the n table of a N->1 relation
	Modifed	Form event and On data change
	MODIFY SUBRECORD	MODIFY RECORD in the n table of a N->1 relation
Form events	Activated After Before Deactivated During In break In footer In header Outside call	Replace with Form event and the appropriate event
Graphs	GRAPH using a 4D Graph Area	Use a SVG picture instead
	GRAPH TABLE	Build the data in arrays and call GRAPH in a SVG picture, or see 4D Chart deprecation chapter
Hierarchical lists	REDRAW LIST	Remove in code (does nothing since v11)
Object properties	DISABLE / ENABLE BUTTON	OBJECT SET ENABLED
Pictures	PICTURE TYPE LIST	PICTURE CODEC LIST
	QT COMPRESS PICTURE	CONVERT PICTURE
	QT COMPRESS PICTURE FILE	WRITE PICTURE FILE / PICTURE TO BLOB
	QT LOAD COMPRESS PICTURE FROM FILE	READ PICTURE FILE / CONVERT PICTURE
	SAVE PICTURE TO FIL	WRITE PICTURE FILE
Resources	(all commands)	See “Mac resources” deprecation chapter
SQL	USE EXTERNAL DATABASE	SQL LOGIN
	USE INTERNAL DATABASE	SQL LOGOUT
String	Convert case ISO to Mac Mac to ISO Mac to Win Win to Mac	CONVERT FROM TEXT / Convert to text when necessary. Just remove the command from the method if conversion is not necessary (which means “the database runs in Unicode mode”)

Subrecords	(all commands)	See “Subtables” deprecation chapter Replace “nnn SUBRECORD” and “nnnSUBSELECTION” with an action on the N record or N-selection of the N-table in a N->1 relation
System documents	Document type	See documentation
User interface	Get platform interface SET PLATFORM INTERFACE	See “Old look” deprecation chapter
Web server	Commands: SET WEB DISPLAY LIMITS SET WEB TIMEOUT Web context Others: Magic **4DBLANK** url **4DMETHOD** html tag Web Conversion Mode selector in GET/SET DATABASE PARAMETER AP SET WEB FILTERS (since v11, the conversion of characters is much better) Web Server functionalities: • 4D HTTP Server as CGI-application using 4DISAPI.DLL and NPH-CGI4D.exe • JavaScript encapsulation Anything related to WebStar (4D Link, 4D Connect, 4D SSI, Welcome, WEBINCLUDE)	See “Contextual mode” deprecation chapter

4D SAS
Worldwide Headquarters
60, rue d'Alsace - 92110 Clichy - France
Tel: +33 1 40 87 92 84 - Fax : +33 1 40 87 92 01
info.intl@4d.com - www.4D.com

4D SAS
France
+33 1 40 87 92 00
info-fr@4d.com

4D, Inc
USA & Canada
+1 800 785 3303
info@4d.com

4D Deutschland GmbH
Germany & Austria
+49 8165 95 19 0
info-de@4d.com

4D Japan
Japan
+81 3 5433 3461
info-jp@4d.com

4D UK Ltd
United Kingdom
+44 1625 536 178
info-uk@4d.com

4D Hispano
Spain, Portugal & Latin America
+34 91 414 92 90
info-es@4d.com
info-pt@4d.com
info-br@4d.com

4D Sweden AB
Sweden & Denmark
+46 8 750 63 00
info-se@4d.com

4D Australasia
Australia & New Zealand
+61 2 9499 9544
info-au@4d.com