4D v18.x Release Notes

Welcome to 4D v18. You will find below several links to pages providing information about new and modified features in this release.

- **What's new in 4D v18** - A blog post that lists all new features and enhancements in 4D v18.
- **Fixed bugs list for 4D v18.2** - The list of all bugs that have been fixed in 4D v18.2.
- **Fixed bugs list for 4D v18.1** - The list of all bugs that have been fixed in 4D v18.1.
- **Fixed bugs list for 4D v18** - The list of all bugs that have been fixed in 4D v18.
- **Changes and updates** - A list of modifications in the new release that may impact your existing developements.
- **Deprecated or removed features** - A list of features, settings, commands, etc., that have been marked as **deprecated** (for future removal) or that have been **removed** in the current release.

Notes:

- Refer to the **Conversion to 4D v18** dedicated manual if you consider converting your database in 4D v18.
- Refer to **Converting databases to projects** in the *4D Design Reference* if you consider converting a binary database to a project database.

Changes and updates

During the R-release cycle, various libraries and components used by 4D are updated, existing behaviors are modified, known issues are recorded. This information, initially published via the "Release Notes" of each version, is summarized on this page.

4D applications

4DRequestsLog format

- **4DRequestsLog files format was modified as of 4D v17R5:**
  - 3 additional fields have been added
  - durations are now in microseconds instead of milliseconds

If you use log analysis tools, you may need to update them.

Settings files automatically moved

With respect to the evolution of settings file architecture, 4D will automatically rename and move some default configuration files:

- "BuildApp.xml" is renamed "buildApp.4DSettings" and moved from the "Preferences" folder to the database "Settings" folder
- "Backup.xml" is renamed "backup.4DSettings" and moved from the "Preferences" folder to the database "Settings" folder
- If the resulting Preferences folder is empty, it is deleted.

Current Logs folder location

- **Logs folder location has been modified:**
  - In v18 and higher versions, the "current" Logs folder is created by default at the same level as the data file. A specific Logs folder is created in the user's current system folder (4D folder) for maintenance logs (compact, verify, repair)
  - Existing Logs folders are left untouched during database conversion.

Vertical scaling operator

Because of the support for inline or multiline comments /*...*/, the picture vertical scaling operator is modified in 4D v18 and higher.

- former operator: */
- new operator: *|

The former operator is automatically replaced when the database is converted to 4D v18 or higher.

Range checking

As of 4D v17 R6, range checking is always enabled in 4D. Consequently, on the "Compiler" page of the Database Settings dialog box, the **Range Checking** checkbox is removed.

Range checking special comments are still supported:

```
// %R- to disable range checking
// %R+ to enable range checking
// %R* to reset to default value (as set in the Settings dialog box)
```

Note that the // %R* comment, which could restore a "disabled" state if the checkbox was unchecked in the Database Settings, is now always equivalent to a // %R+ (option enabled).

On Timer no longer triggered when resizing a form

Because the On Resize event is now triggered when a subform is resized, the On Timer form event is no longer

triggered in this case to prevent conflicts. Previously, this event could be triggered in but with a non-reliable frequency.

Form event command renamed

The previous **Form event** command has been renamed **Form event code** and a new **FORM Event** command, returning an object, has been implemented.

Better detection of Blob parameters

The compiler has been enhanced to better detect a mismatch when using Blob variables. The compiler will display an error when a command requires a Blob but receives a non-Blob parameter.

Indexed object fields might need reindexing

Depending on stored data and how it was created, an index of object fields could be incorrect. This is an extremely rare, unreproducible situation. One factor in this situation appears to have been objects with special characters (e.g.: .,'",[]) in the property names. If this occurs, the index may become corrupted and cause query by attribute in object fields to fail (ACI0098796). This concerns databases created with 4D v15 - v17.0 and with indexed object fields. Databases created or indexed with v16.4 HF1 / v17.1 / v17 R4 or later are not impacted. To correct this issue, it is recommended to reindex object fields.

New 4D Remote resource cache folder

With 4D v17 R5, it's now possible to run several copies (macOS and Windows) or instances (Windows) of the same application connected to the same server. To allow this, the naming of the cache folder has been changed. The naming in previous versions was based on Server-IP and Server-Port number, now with an additional instance number. The old cache folder and its data, will no longer be used. If you want to delete it, it needs to be removed manually.

4D compiler updated

In order to implement the new **Get call chain** command, the code generated by the 4D compiler has been updated. As a consequence, databases compiled with 4D v17 R6 can only be opened by 4D v17 R6 or higher. Databases compiled with previous versions need to be recompiled if you want to take advantage of the new command.

Formula commands renamed

The "New formula" and "New formula from string" commands have been renamed to **Formula** and **Formula from string**.

New name and parameter for On 4D Mobile Authentication database method

As of 4D v18, the former "On 4D Mobile Authentication" database method has been modified:

- it is renamed **On REST Authentication database method**,
- it accepts a new $4 parameter to receive the ip address of the caller.

4D View Pro

On data Change event not triggered anymore

4D events in 4D View Pro have been refactored, and the event On Data Change is not triggered anymore.

New date format

To better align with the SpreadJS date format, 4D has changed the way dates and times are stored. Since 4D v17 R2, when converting a 4D view to a 4D View Pro document, dates and times were cast as a C_TEXT. As of 4D v17 R4 dates and times are cast as a C_OBJECT which contains 2 fields: one for the date (C_DATE), the other for the time (C_TIME).

Regional settings can be the same as the OS

As of 4D v17 R5, the 4D View Pro default localization settings are the same as the current 4D host database. Thus in 4D v17 R5, your 4D View Pro interface could display date, time, and number differently than in previous 4D versions.

4D Write Pro

Page number and count not frozen

The *$wp_pageNumber* and *$wp_pageCount* expressions are always evaluated and cannot be frozen by the **ST FREEZE EXPRESSIONS** command (same behavior as with the 4D Write plug-in).

Table pagination compatibility

In previous versions, when a range was inside a row and a column, or when a page break was inserted, the break was inserted before the table. Now, the break will be inserted before the row, and the table will be splited into two parts.

New way to manage default tabs

The behavior of wk tab stop offsets has been upgraded. The last element of a tab parameter is no longer used to define the tab offset default value. It now defines the offset value of the last tab. Starting with 4D v17 R5, the default tab value is managed by the wk offset parameter of the wk tab default attribute, which has been created to manage default tab values.

OS Notes

4D Developer Edition blocked on macOS High Sierra (10.13)

When you open 4D Developer Edition v18 or higher on High Sierra, a warning message is displayed: "4D can't be opened because it is from an unidentified developer".

We recommend you to use macOS Mojave or Catalina. However, if using macOS High Sierra for development is a requirement, there are two possible workarounds:

- install Swift from Apple, so the 4D Developer application will be correctly identified.
- or, go to the "System Preferences > Security" dialog box, and click on the "Open anyway" button (pay attention to the fact that this is to be done after each download of a new build).

Note: Only the 4D Developer application is concerned. Other editions (4D Server, merged applications) can be directly opened without any restriction.

Windows 7 and Windows 8 no longer supported

Windows 7 support is stopped by Microsoft as of January 2020. Therefore, 4D is stopping its maintenance as of 4D v17 R5. In parallel we stop to support Windows 8, please upgrade to Windows 10.

Minimum requirements for Windows 7

Windows 7: It is strongly recommended to install all available Microsoft updates (use update dialog in system settings). The KB2999226 update (released 9/2015) is mandatory to avoid errors regarding api-ms-win-crt-rumtime-1-1-0.dll. Microsoft has already stopped Windows 7 support and the extended support (security fixes) will end soon (Jan/2020). We suggest updating to Windows 10.

Localization issue with Date & Time expressions on macOS Mojave (10.14)

When macOS Mojave uses a localization that doesn't exist in 4D, the date and time might be displayed in US format. For example, if the Mac client uses the Italian localization, for which the "it.lproj" doesn't exist, the date will be displayed in a US format.

Library Updates

- **openSSL**: updated to **openSSL 1.1.1d** in 4D v18
- **Xerces**: updated to v3.2.1
- **CEF**: updated to v3626
- **PHP**: updated to v7.3.1 - this is a major upgrade for PHP, with additional features but also possible compatibility issues. More information.
- **ICU**: update to v63.1 - this major upgrade to newer Unicode version forces an automatically rebuild of alphanumeric, text and object indexes.
- **Hunspell**: updated to v1.7.0

Deprecated or removed features

For over 30 years, our main goal has been to enhance our product (using new concepts and technologies) while ensuring the compatibility of 4D applications. As early adopters of 4D know, we have always put a lot of effort into compatibility, and we can find 4D applications all over the world 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:

- 4D must bring new technologies, new APIs, and new paradigms to developers
- OSes change every day, and sometimes deprecate their own old APIs

This is why 4D sometimes needs to tag some commands and features as deprecated, meaning that one day, they will be removed from the language in a future major version.

Letting developers know what is deprecated and what kind of replacement can be used instead makes it much more comfortable for them to implement the change in their code: There is no urgency, no pressure, and the developer has plenty of time to make any necessary changes.

Summary Table

Feature	Replacement	Status in current 4D version
4D & 4D Server **32-bit versions**	4D 64-bit/4D Server 64-bit	Removed.
4D Write	4D Write Pro	Removed
4D View	4D View Pro	Removed
XSLT commands	Use PHP libxslt module or the **PROCESS 4D TAGS** command	Removed
QuickTime support (compatibility option)	Use native formats	Removed
Float field type	Automatically converted to Real	Removed
Non-Unicode mode (converted pre-v11 database)	Move to Unicode	Removed
Pictures in PICT format	Use modern formats; help is provided by **GET PICTURE FORMATS**	Removed
DatabaseOutsidePackage key	none	Removed
Right-to-left features removed	none	Removed
User Forms	**Dynamic Forms**	Deprecated
Dynamic assignment of variables received through HTTP (compatibility option for databases created prior to v13.4)	**WEB GET VARIABLES** command (to recover variables). **WEB GET BODY PART**/ **WEB Get body part count** commands (to recover posted files)	Deprecated
Converted **Subtables**	Use N->1 tables	Deprecated
Mac OS QuickDraw fonts no longer supported	Font names	Removed
API QuickDraw for plug-ins	New SDK plug-in for third-party plug-ins	Removed
File and folder APIs on Mac	**Path to object**, **Object to path**	Deprecated
Code-based standard actions	String patterns ("ak" constants in the **Standard Action** theme)	Deprecated
Mac Resources	Use "Resources" folder. For compatibility, you can still use it in converted databases. We no longer support write access commands.	OS (cicn icons: removed.) Database .RSR files are not opened automatically. You need to use **Open resource file**.
Management of CPU priority	none	Deprecated
Legacy network layer	Use ServerNet	Deprecated

4D Mobile	ORDA and REST Server	Deprecated

Explanation of values for the “Status” column:

- **Removed**: No longer available in the current version (or the version indicated).
- **Deprecated**: Should no longer be used and will be removed in a future major version.
- **OS**: Depends on officially deprecated OS technologies (*e.g.*: PICT format). Status is the same as Deprecated, but an OS could remove the support before we do.

Language: deprecated commands

Every obsolete command is prefixed by "_o_" and is not available in 4D lists (code editor, type-ahead feature, etc.). Obsolete (or deprecated) commands will not be removed from existing code and will continue to work normally as long as they are supported. It is still possible (but not recommended) to add an obsolete command in a method by simply entering its name prefixed by "_o_"; it will be interpreted correctly.

Command	Replaced with	Obsolete since
4D Environment theme:		
_o_DATA SEGMENT LIST	-	v11
Backup theme:		
_o_INTEGRATE LOG FILE	**INTEGRATE MIRROR LOG FILE**	v16
Compiler theme:		
_o_ARRAY STRING	**ARRAY TEXT**	v12
_o_C_GRAPH	(use SVG with the **GRAPH** command)	v12
_o_C_INTEGER	**C_LONGINT**	v12
_o_C_STRING	**C_TEXT** (as soon as database is in Unicode)	v12
Data Entry theme:		
_o_ADD SUBRECORD	**ADD RECORD** in the n table of a N->1 relation	v12
_o_MODIFY SUBRECORD	**MODIFY RECORD** in the n table of a N->1 relation	v12
Drag and Drop theme:		
_o_DRAG AND DROP PROPERTIES	Commands from the **Pasteboard** theme	**v17 R4**
Form Events theme:		
_o_During	Replace with **Form event code** and the appropriate event	v12
Forms theme:		
_o_FORM GET PARAMETER	Disabled	**v17 R4**
Graphs theme:		
GRAPH (using 4D Graph Area)	Use an SVG picture instead	v12
Hierarchical Lists theme:		
_o_REDRAW LIST	Remove in code (does nothing since v11)	v11
Language theme:		
_o_NO TRACE	Use the No trace button of the debugger	**v18**
List Box theme:		
LISTBOX Get property constants:		

_o_lk display hor scrollbar/ _o_lk display ver scrollbar	**OBJECT GET SCROLLBAR**	v16 R3
_o_lk footer height	**LISTBOX Get footers height**	v16 R3
_o_lk header height	**LISTBOX Get headers height**	v16 R3
_o_lk hor scrollbar position/ _o_lk ver scrollbar position	**OBJECT GET SCROLL POSITION**	v16 R3
Objects (Forms) theme:		
_o_DISABLE BUTTON/ **_o_ENABLE BUTTON**	**OBJECT SET ENABLED**	v12
_o_OBJECT Get action	**OBJECT Get action**	v16 R3
_o_OBJECT SET COLOR	**OBJECT SET RGB COLORS**	**v18**
Pictures theme:		
_o_PICTURE TO GIF	**PICTURE TO BLOB**	v16 R5
Printing theme:		
_o_PAGE SETUP	**SET PRINT OPTION**, **GET PRINT OPTION**, **Print settings to BLOB**, **BLOB to print settings**	**v18**
Selection theme:		
_o_MOBILE Return selection	ORDA	**v18**
SQL theme:		
_o_USE EXTERNAL DATABASE	**SQL LOGIN**	v12
_o_USE INTERNAL DATABASE	**SQL LOGOUT**	v12
String theme:		
_o_Mac to Win		v11
_o_Win to Mac		v11
Subrecords theme: all commands	Replace “nnn SUBRECORD” and “nnn SUBSELECTION” with an action on the N record or N-selection of the N-table in a N->1 relation	v12
System Documents theme:		
_o_Document type	**Path to object**	v12
_o_Document creator	**Path to object**	v16 R6
_o_SET DOCUMENT TYPE	**Object to path**	v16 R6
_o_SET DOCUMENT CREATOR	**Object to path**	v16 R6
_o_MAP FILE TYPES	Use UTIs and Info.plist	v16 R6
System Environment theme:		
_o_Font name	Use font identifiers The **OBJECT SET FONT** command no longer accepts a LongInt parameter for the font: this parameter is now a String and you must specify the font name.	v14
_o_Gestalt	**Get system info** / **Is macOS** / **Is Windows**	v17
_o_PLATFORM PROPERTIES	**Get system info** / **Is macOS** / **Is Windows**	v17
User Forms theme:		

_o_CREATE USER FORM	Dynamic Forms	**v17 R4**
_o_DELETE USER FORM	Dynamic Forms	**v17 R4**
_o_EDIT FORM	Dynamic Forms	**v17 R4**
_o_LIST USER FORMS	Dynamic Forms	**v17 R4**
4D Internet Commands:		
FTP_Progress	Returns an error if called Note: *progress* parameter is no longer supported with **FTP_Append**, **FTP_Receive**, **FTP_Send**	v16 R2

You can find an alphabetical list of obsolete commands in the **Deprecated Commands** appendix of the **4D Language Reference**.

Language: removed commands

The following commands, previously deprecated, have been removed from 4D and must not be used in your code. If they are called, they will trigger an error (*Error 33 - Unimplemented command or function*).

Warning: Calls to removed commands cannot be detected by the check syntax or compile features. You must verify your code to make sure that it does not contain any calls to removed commands.

Command	Replaced with	Removed in
4D Environment theme:		
_o_ADD DATA SEGMENT	-	v17 R5
Graphs theme:		
_o_GRAPH TABLE	Build the data in arrays and call **GRAPH** in a SVG picture	v17 R5
Pictures theme:		
_o_PICTURE TYPE LIST	**PICTURE CODEC LIST**	v17 R5
_o_QT COMPRESS PICTURE	**CONVERT PICTURE**	v17 R5
_o_QT COMPRESS PICTURE FILE	**WRITE PICTURE FILE**/ **PICTURE TO BLOB**	v17 R5
_o_QT LOAD COMPRESS PICTURE FROM FILE	**READ PICTURE FILE**/ **CONVERT PICTURE**	v17 R5
_o_SAVE PICTURE TO FILE	**WRITE PICTURE FILE**	v17 R5
Resources theme:		
_o_ARRAY TO STRING LIST	-	v17 R5
_o_Create resource file	-	v17 R5
_o_DELETE RESOURCE	-	v17 R5
_o_Get component resource ID	-	v17 R5
GET ICON RESOURCE	-	v17 R5
_o_SET PICTURE RESOURCE	-	v17 R5
_o_SET RESOURCE	-	v17 R5
_o_SET RESOURCE NAME	-	v17 R5
_o_SET RESOURCE PROPERTIES	-	v17 R5
_o_SET STRING RESOURCE	-	v17 R5
_o_SET TEXT RESOURCE	-	v17 R5
String theme:		
_o_Convert case	**CONVERT FROM TEXT**/ **Convert to text** when necessary.	v17 R5
_o_ISO to Mac	Just remove the command from the method if conversion is not necessary	v17 R5
_o_Mac to ISO	(which means the database runs in Unicode mode)	v17 R5
System Environment theme:		

_o_Font number	Use font identifiers	v17 R5
User Interface theme:		
_o_Get platform interface/_o_SET PLATFORM INTERFACE	Can be used only for converted application; with the Automatic Platform constant	v17 R5
_o_INVERT BACKGROUND	-	v17 R5
Web server theme:		
_o_SET CGI EXECUTABLE	-	v17 R5
_o_SET WEB DISPLAY LIMITS	-	v17 R5
_o_SET WEB TIMEOUT	-	v17 R5
_o_Web Context	-	v17 R5
Windows theme:		
_o_Open external window	-	v17 R5
XML theme:		
_o_XSLT APPLY TRANSFORMATION	Use PHP *libxslt* module or the **PROCESS 4D TAGS** command	v17 R5
_o_XSLT GET ERROR	Use PHP *libxslt* module or the **PROCESS 4D TAGS** command	v17 R5
_o_XSLT SET PARAMETER	Use PHP *libxslt* module or the **PROCESS 4D TAGS** command	v17 R5

Legacy technologies (project compliance)

Projects represent an important evolution of 4D databases architecture. Since projects rely on latest software interface requirements and most modern technologies, they do not support some legacy features, listed below. These features are now deprecated and should no longer be used in your databases, for a better compliance with the project architecture.

For detailed information on the project architecture in 4D, please refer to the 4D projects documentation on developer.4d.com. 4D provides an automatic export tool to convert .4db databases to projects. During the export, deprecated features are automatically converted, removed or generate errors (see **Converting databases to projects**).

Form objects and properties

The following form objects and properties do not comply with current interface requirements and are now deprecated. They are not supported in **Dynamic Forms**, and may generate a warning or an error in the project conversion log file (see comments).

Deprecated feature	Conversion status	Comment
Highlight buttons	error	Must be converted to 3D buttons
Picture radio buttons	error	Must be converted to 3D buttons
Dials	error	Must be converted to progress indicators
Matrix	warning	Matrix objects are automatically converted to svg pictures and stored in the resources folder of the database
Boolean field as radio buttons	warning	Supported but automatically converted to a pair of standard grouped radio buttons with associated expressions: *[table]Boolean_field* and *Not([table]Boolean_field)*
On Background picture format	-	Converted to Truncated (non-centered)
List box - Scrollable area compatibility	warning/error	Use regular list box features
List box - Connected list boxes compatibility	error	Use standard list box features

Platform interface "printing" property	warning	Objects with "printing" property are automatically converted to "flat" style (button, checkbox, radio button, variable/field with "system" border)

Database structure options

The following database structure options are deprecated and will be edited or generate errors in the project conversion log file (see comments).

Deprecated feature	Conversion status	Comment
"Can't Modify" field option	warning	Automatically moved at form level during export to project
"Display only" field option	warning	Automatically moved at form level during export to project
"Mandatory" field option	error	Select "Reject NULL value input" option

Toolbox

The following Toolbox editors or features are deprecated and are not supported in projects:

Deprecated feature	Conversion status	Comment
Picture library	warning	Pictures are automatically exported to the resources folder of the database
GET PICTURE FROM LIBRARY	-	Do not work - Use **READ PICTURE FILE** instead
"Editable by user" list option	-	
LIST OF CHOICE LISTS	-	-
SAVE LIST	-	Error at runtime if called from a project
Group named "" or "*"	error	Reserved in projects
Standard user named "Designer" or "Administrator"	-	Names reserved in projects

Compatibility settings

Legacy compatibility modes are not supported in projects. Compatibility settings are reset as for a new database during the export to project. If your database still rely on old compatibility settings, you need to consider updating it.

We highly recommend that you read the series of blog posts devoted to how to get rid of compatibility settings

Removed or deprecated as of 4D v18

32-bit versions

Starting with 4D v17 R5, 32-bit versions of 4D and 4D Server are no longer produced or delivered. Only 64-bit versions are available, allowing the 4D applications to take full advantage of this powerful architecture.

Although 64-bit versions support most legacy 4D features, they favor most recent technologies and do not support those that have been declared obsolete in previous versions. Upgrading databases developed for 32-bit versions may require some adaptations. Please consult the "Changing from 32-bit versions to 64-bit versions" page in the *Conversion to v17* manual.

XSLT commands

XSLT is no longer available in 4D. Calling an XSLT command will generate an error 33 "Unimplemented command or function".

To support our customers still using XSLT in 4D, we made the choice to rely on the PHP XSL library, which provides a comprehensive API allowing you to perform all operations necessary for your XSL transformations. This library is an efficient tool which can easily replace the removed **_o_XSLT APPLY TRANSFORMATION**, **_o_XSLT SET PARAMETER** and **_o_XSLT GET ERROR**. 4D has produced a specific document to help you use PHP XSL as a replacement for the 4D XSLT commands: Download XSLT with PHP technical document (PDF).

We also suggest that you consider using 4D tags when dealing with the dynamic generation of HTML pages, since in most cases it is easier if you handle HTML code as unformatted text (see also the **PROCESS 4D TAGS** command).

QuickTime

QuickTime

Support for QuickTime (through a database parameter) is removed from 4D starting with v17 R5. Picture codecs related to QuickTime are no longer supported on macOS and Windows.

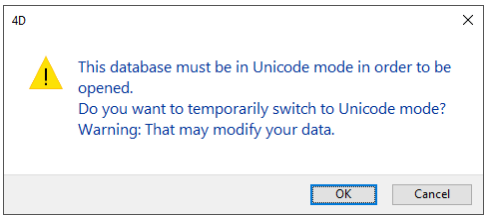
Float field type

As of v18, Float data field type is no longer supported the 4D database. This field type was only used by the SQL engine of 4D. If your database was using a Float type field, it is automatically converted to Real type when the database is open in v18 or higher. Existing data are not converted, unless you compact the data or re-save records.

Note that this data type was only fully handled by the SQL engine. Float values were converted to Real values in 4D if they were handled via the 4D language.

Non-Unicode mode

Support for ASCII mode (synonym for “non-Unicode mode”) is removed from 4D v17 R5 and higher. When opening a converted database that was not set to Unicode, 4D displays the following alert dialog box:



This alert will be displayed at each startup while the **Unicode mode** option is not checked in the "Compatibility" page of the database settings. You must – for converted structures – activate the Unicode mode. The Conversion to 4D v14 PDF document gives hints about this topic.

Pictures in PICT format

The PICT format (deprecated by Apple since 2005) is no longer supported in 4D starting with v17 R5. It means that pictures in PICT format cannot be rendered by any means, they must be converted. Use the **GET PICTURE FORMATS** command to detect and filter pictures using the PICT format in your data file.

DatabaseOutsidePackage key

Specific deployment mode for merged single-user 4D applications on the mac platform using a *DatabaseOutsidePackage* key in the *info.plist* is no longer supported. Starting with 4D v17 R5, the *DatabaseOutsidePackage* key is ignored.

Right-to-left features removed

As for 4D v17 R4, 4D no longer provides specific support for right-to-left interfaces on Windows. Corresponding options are no longer available in the database settings or form properties. In converted databases, right-to-left settings will be ignored at runtime.

User Forms

As of 4D v17 R4, user forms are deprecated. To provide customizable user interfaces, it is now recommended to use **dynamic forms** (see **Dynamic Forms** in the **4D Design Reference**). All commands within the "User Forms" theme have been deprecated.

User forms are still supported for compatibility, however, they should no longer be used. Their support will be removed in future releases.

4D Mobile

4D Mobile feature is deprecated as of 4D v18 and should no longer be used. Access to 4D data through REST requests is now supported by ORDA technology and the 4D REST Server.

Already deprecated in previous releases

Dynamic assignment of variables received through HTTP

In previous versions of 4D, the Web server automatically recopied the value of variables sent through a Web form or a URL into 4D variables when they had the same name.

For reasons of optimization and control, this principle is not maintained starting with 4D v14: the value of Web

variables are no longer automatically assigned to the 4D variables. To recover variables sent using a POST or a GET, you must use the **WEB GET VARIABLES** command exclusively. To recover the posted files, you must use the **WEB GET BODY PART**/ **WEB Get body part count** commands.

Note: Dynamic assignment is also disabled by default in 4D databases created beginning with version 13.4.

However, for compatibility, this mechanism is maintained by default in databases created with a version of 4D earlier than 13.4. In this case, you can disable it using the **Automatic variable assignment** compatibility option on the Compatibility page of the Database Settings.

Since this mechanism is obsolete, we strongly recommend that you uncheck this option in your converted databases (and adapt your code if necessary) so as to facilitate future evolutions.

Subtables

Over several major versions, 4D has warned developers against the use of subtables and since 4D v11, it is no longer possible to create a field of the SubTable type. Subrecords have several known limitations. For example, they are always loaded in memory; they are not handled by the **SEND RECORD** or **DUPLICATE RECORD** commands.

We do not 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 do plan to remove them in a future major version of 4D. Developers who used subtables for performance reasons (certain specific situations where loading related records was slow) can be reassured, especially since 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 tech tip; just a quick overview):

- Before conversion from a pre-v11 structure: in 2004, create the appropriate N table and the ID field in Table 1 (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 allows 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 if it is needed (see below).

What we mean by "change the code everywhere if 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 **_o_CREATE SUBRECORD** with **CREATE RECORD**, filling the ID fields)
  - Explicitly load the N records when needed

Note: Starting with 4D v14 R3, you can assign values to the special "id_added_by_converter" fields that are automatically added by 4D when it converts a database containing subtables. This allows you to keep the "subtable relation" link, and add or modifiy related records, without needing to use deprecated commands such as **_o_CREATE SUBRECORD**. Once you have updated your methods, these special relations can be replaced with standard ones with no change in your code.

Mac OS QuickDraw fonts no longer supported

QuickDraw fonts (*e.g.* Geneva, Chicago) are now deprecated and you should no longer use ID numbers to designate fonts. The **_o_Font name** command is kept for compatibility but will be removed in subsequent versions. The **OBJECT SET FONT** command now only accepts font names.

API QuickDraw for plug-ins

There are two types of plug-ins: those using the new plug-in API, and those that still use the old one (with QuickDraw).

For plug-ins using the old tool box (with QuickDraw): to maintain compatibility, the drawing/rendering is no longer done directly in a QuickDraw port, as in previous versions, but instead through a GWorld QuickDraw offscreen area dedicated to the plugin.

Consequently, you have to respect a few rules, like plugins must not modify the current port set by the container (form object).

For plug-ins using the new tool box: only this new tool box is used and not QuickDraw (see https://github.com/4D/4D-Plugin-SDK).

File and folder APIs on Mac

Starting with 4D v16 R6, Mac OS 9-based APIs for file and folder management are deprecated in 4D. These APIs have already been deprecated by Apple for a long time.

4D commands relying on the old APIs have been renamed: **_o_Document type**, **_o_Document creator**, **_o_SET DOCUMENT TYPE**, **_o_SET DOCUMENT CREATOR**, and **_o_MAP FILE TYPES**. These commands are maintained for compatibility, but are deprecated and should no longer be used. 4D includes two new commands to handle pathnames and file extensions: **Path to object** and **Object to path**.

Code-based standard actions

Standard actions have been internally redesigned in 4D v16 R3. They are now based on string patterns and support parameters (for more information, please refer to the **Standard actions** section).

In your converted applications, standard actions are transparently remapped to the new design. However, in the 4D language, standard actions were previously based on code values. This principle is deprecated and should no longer be used:

- Code-based text standard actions in the **Standard Action** constant theme (previously named "Text Values for Associated Standard Action") are prefixed with "_o_" and should no longer be used.
- The whole **Value for Associated Standard Action** constant theme is deprecated.
- The **_o_OBJECT Get action** command (which returns a code) is deprecated and has been replaced by the updated **OBJECT Get action** command (which returns a name).

Mac Resources

This is another old Mac OS technology, deprecated since Mac OS X 10.4 (Tiger, 2005). Resources are used to store structured data such as text and strings (localization), as well as icons, etc. Basically, we can say that it's not the resources that are deprecated, it's their on-disk support, known as the resource fork. The resource fork is part of the Mac OS file system, and since the beginning of Mac OS X, Apple has tried to remove this support as it is not compatible with other file systems (Unix, Windows), and is the source of a lot of problems when files are 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 the " **Pictures in PICT format**" topic above.)
- The count of resources and the size of the resources are limited (about 2,700 resources or 16 MB)

We have removed 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
  - Save any private resources as appropriate for you (typically: save as a binary file with a specific extension)
- Use the "Resources" folder to store your resources. Use **Get 4D folder**(Current resources folder) to dynamically

get the parent path for your resources.

Management of CPU priority

Modifying the CPU priority is an obsolete setting. Memory management for 4D applications has been optimized over the different versions and it is unnecessary and may even be counterproductive to modify the default settings. As a result:

- the "CPU Priority" area ("General" page of the Database Settings) is now displayed only in converted databases where a custom value has previously been set. We recommend clicking on the **Factory settings** button in this case in order to reinitialize these parameters and remove them from the dialog box.

- beginning with 4D v16 R2, obsolete selectors of the **SET DATABASE PARAMETER** command have been renamed (_o_4D Remote mode scheduler, _o_4D Local mode scheduler and _o_4D Server scheduler) and must no longer be used.

Previous Documents

This document concerns the **4D v18 product range**. For reference, you can consult previous documents (PDF) describing deprecated features in prior product ranges, available here:

- Deprecated and Removed Features in 4D v17 - **(Rev. Feb 2018)**
- Deprecated and Removed Features in 4D v16 - **(Rev. Jan 2017)**
- Deprecated and Removed Features in 4D v15 - **(Rev. June 2015)**
- Deprecated and Removed Features in 4D v14 - **(Rev. Oct 2014)**
- Deprecated and Removed Features in 4D v13 - **(Rev. 20 Feb 2012)**
- Deprecated and Removed Features in 4D v12 - **(Rev. 03 Jun 2010)**