

# Conversión a 4D v17

---

Bienvenido a este manual "Conversión a 4D v17", que describe varios puntos que se deben verificar antes, durante y después de la conversión de una base 4D v16 a 4D v17.










La conversión de las bases 4D v16 debe realizarse sin problemas a 4D v17, y le ofrecemos algunas recomendaciones para asegurarnos de que todo salga bien en el capítulo "**Principios para la conversión**". Sin embargo, una vez que se realiza la conversión, hay un par de cosas que debe verificar, como "**Nuevas opciones de compatibilidad**" y "**Cambios en el comportamiento**" tanto a nivel de la aplicación como en los comandos 4D, que debe comprender para hacer el mejor uso de las novedades 4D v17.


Y finalmente, este manual resume las **funcionalidades obsoletas de 4D v17** que los desarrolladores necesitan detectar rápidamente para evaluar el tiempo necesario para configurar nuevas funciones.


**Nota:** algunas modificaciones descritas en este manual han sido introducidas durante el programa 4D v16 "R-release".

Para la conversión de base anteriores o incluso mucho más antiguas, a menudo es necesario utilizar versiones intermedias. Y para los diferentes puntos a comprobar, consulte los documentos de conversión de las versiones anteriores:

- **4D v16:** "[Conversión a 4D v16](#)" y "[Funcionalidades obsoletas 4D v16 y superior](#)".
- **4D v15:** "[Conversión a 4D v15](#)" y "[Funcionalidades obsoletas 4D v15 y superiores](#)".
- **4D v14:** "[Conversión a 4D v14](#)" y "[Funcionalidades obsoletas 4D v14 y superiores](#)".
- **4D v13:** "[Conversión a 4D v13](#)" y "[Funcionalidades obsoletas 4D v13 y superiores](#)".
- **4D v12:** "[Funcionalidades obsoletas 4D v12 y superiores](#)" (no había documento "Conversión" para esta versión).
- **4D v11:** "[Conversión a 4D v11 SQL](#)".

-  Principios para la conversión
-  Diálogo Compatibilidad
-  Cambios de comportamiento
-  Cambios de nombre o de tema
-  Funcionalidades obsoletas
-  Funcionalidades desactivadas
-  Cambiar de 32 bits a 64 bits
-  Convertir los documentos 4D Write en 4D Write Pro
-  Conversión de documentos 4D View a 4D View Pro

 Anexo: Notas del lanzamiento 4D v16 Rx

 Anexo: métodos útiles para la conversión

# Principios para la conversión

## Qué hacer antes de convertir

---

- Debe tener una versión "interpretada" de la base (archivo xxxx.4DB para la estructura, archivo xxxx.4DD para los datos, archivos .RSR y .4DR para las versiones anteriores a 4D v11, ver los documentos "Conversión" de las versiones anteriores), así como también la contraseña **Diseñador** para realizar una conversión;
- Haga una **copia de su base** antes de la conversión;
- Realice una verificación de sintaxis. Incluso si no desea compilar su base, esta verificación puede ayudarle a alertarle sobre posibles errores;
- Utilice el **Centro de Mantenimiento y Seguridad** para verificar y reparar la estructura y los datos;
- Compruebe si tiene algún PICT utilizando el comando **GET PICTURE FORMATS** y conviértalos utilizando el comando **CONVERT PICTURE**. En 4D a partir de la v14, por defecto los codecs QuickTime no son soportados. Por compatibilidad, en una versión 32 bits, puede reactivarlos en su base con la ayuda de un selector QuickTime Support para el comando **SET DATABASE PARAMETER**); La modificación de esta opción necesita del reinicio de la base. Sin embargo, tenga en cuenta que el soporte con QuickTime se eliminará permanentemente en futuras versiones de 4D.  
Para convertir imágenes de formato obsoleto: ver **Conversión de imágenes en estructura**.
- (Opcional) Posibilidad de implementar llaves primarias si se necesita el registro de datos (a partir de la versión 14) (ver **Definir o eliminar una llave primaria** en el manual *Diseño*). Se recomienda configurarlos, pero esto se puede hacer después de la conversión.
- Desde la versión 13.5, es obligatorio que los campos **Únicos** sean **indexados**. Ya no se permitirá crear/modificar registros para un campo único no indexado: al intentar guardar el registro se generará un error (-9998 registro único existe, el índice 1088 no es válido o falta).  
Para crear índices faltantes o generar un archivo disco que liste los campos no indexados, consulte **Para crear los índices faltantes**.

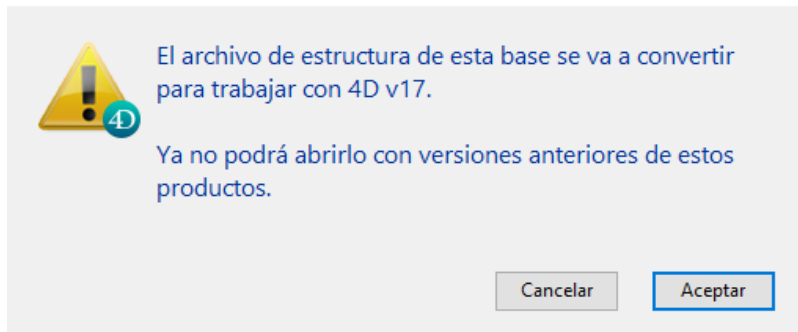
## Cómo convertir

---

Las bases de datos creadas con la versión 16 de 4D o 4D Server (así como también las creadas en v11, v12, v13, v14 y v15) son compatibles con 4D versión 17 (archivos Estructura y datos). Puede convertir cualquier archivo de estructura interpretado. Para ello, solo lance 4D v17 y abra su archivo de Estructura en modo interpretado (archivo xxx.4DB).

- **Conversión del archivo de estructura (.4DB)**

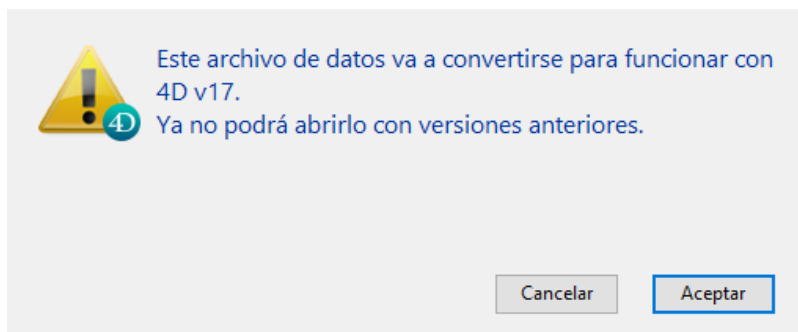
Un diálogo le informa sobre la conversión del archivo de estructura y, dependiendo de la versión de inicio, de la conversión del archivo de datos:



Su archivo de estructura se convierte a 4D v17 y no se puede abrir en una versión anterior.

- **Conversión del archivo de datos (.4DD)**

Los datos se deben convertir para 4D v14 y bases de datos anteriores. En este caso, aparece un segundo diálogo:

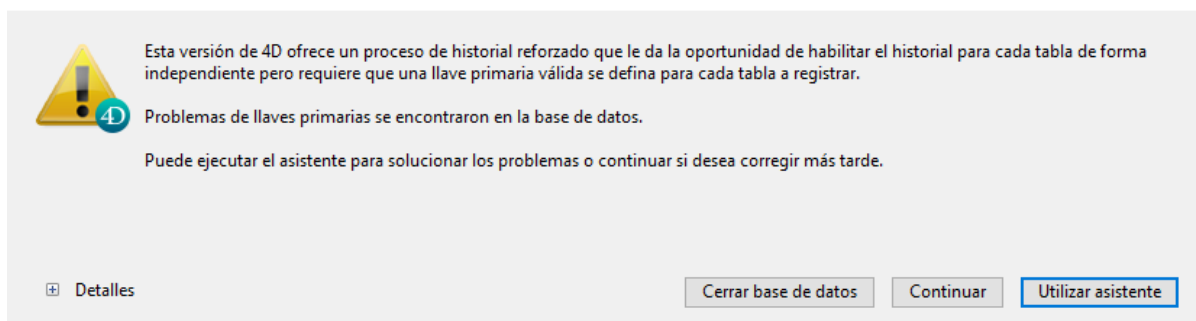


Este archivo de datos se convierte a la versión 17, pero aún se puede abrir y usar con 4D v15 o 4D v16.

No es necesario convertir los datos para las bases 4D v15 o 4D v16.

- **Diálogo de configuración del archivo de historial**

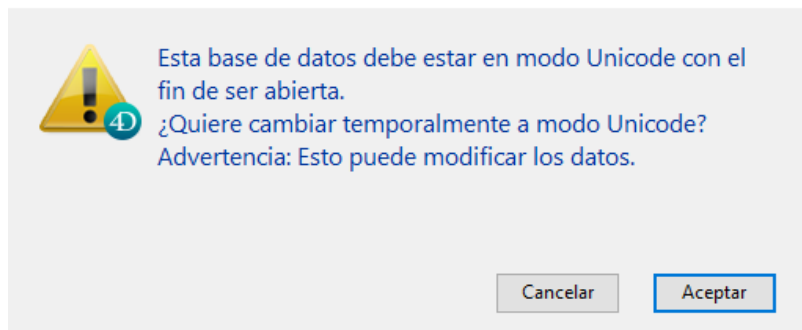
Si las llaves primarias no están en su lugar, 4D le pedirá que lo haga mostrando el siguiente diálogo:



Es aconsejable configurar las llaves primarias (botón "Usar el asistente"), pero este paso se puede hacer más adelante (botón "Continuar"), para las tablas que requieren el registro (para una copia de seguridad). Para más información, consulte el capítulo **Gestión de llaves primarias**.

- **Diálogo Unicode temporal**

Si abre su aplicación en 4D v17 64 bits y su aplicación de inicio no está en modo Unicode, 4D le ofrecerá cambiar temporalmente su base a Unicode.



Como Unicode mejora la velocidad, este paso debería haberse realizado desde varias versiones. Si este no es el caso, hágalo rápidamente. Para más información, consulte el capítulo [Página Compatibilidad](#).

## Después de la conversión

---

Utilice el **Centro de mantenimiento y seguridad** (CSM) para verificar y reparar la estructura y los datos.

Como recordatorio, en la **estructura**:

- detección de métodos huérfanos (`__Orphan__xxxxx`) se indican por **advertencias** en el archivo de historial del CSM y pueden eliminarse utilizando el Explorador (después de comprobar que su código ya no es útil);
- no se permite detección de nombres de objetos duplicados en formularios: se señalan como **advertencias** en el archivo de historial del CSM. Puede realizar una operación de reparación en la base para modificar estos nombres (en este caso, asegúrese de revisar la programación de los nombres de objetos).
- Detección de imágenes obsoletas (formato PICT). Ver [Verificar la aplicación](#) en el capítulo CSM.  
Ver el párrafo [Imágenes en formato PICT](#) en el manual de *funcionalidades Obsoletas o eliminadas*. Estas advertencias pueden afectar tanto a las imágenes estáticas como a las almacenadas en la librería de imágenes o en objetos de formulario.

**Nota:** para convertir estas imágenes: ver [Conversión de imágenes en estructura](#).

- **Caracteres indeseables en los nombres (".", "[", and "]")**  
A partir de 4D v16 R4, el uso de puntos (.) Y / o corchetes ([ ]) no se recomienda en los siguientes elementos:
  - nombres de variable
  - nombres de tabla
  - nombres de campo
  - nombre de método proyecto

Para ayudar a los desarrolladores a poner sus aplicaciones conforme con esta regla, la acción [Verificar la aplicación](#) busca automáticamente la presencia de estos caracteres no deseados en los nombres de variables, tablas, campos y métodos. Si se detectan estos caracteres, el CSM informa sobre "anomalías" y el archivo de historial contiene las advertencias apropiadas:

En este caso, se recomienda cambiar el nombre de estos elementos en su aplicación.

### Información sobre los **datos**:

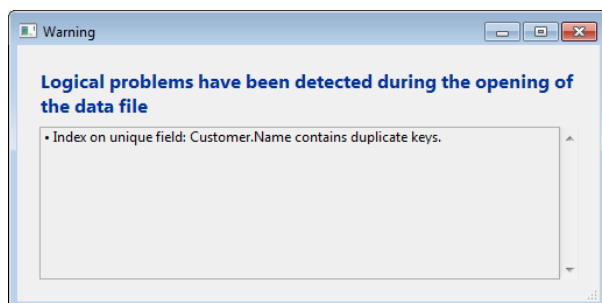
- detección de duplicados en campos únicos:

Se ofrece información adicional: When using the MSC or a command like [#cmd id="939"/], the generated log files now contain the names of the tables and fields involved, as well as each duplicate value.

Cuando se utiliza el CSM o un comando como **VERIFY DATA FILE**, los archivos de historial generados ahora incluyen los nombres de las tablas y campos en cuestión, así como también cada valor duplicado.

**Nota:** al ingresar datos, el cuadro de diálogo de error "Llave duplicada" ahora incluye los nombres de las tablas y campos correspondientes, así como también el valor duplicado y el comando **GET LAST ERROR STACK** también incluye información detallada sobre cualquier duplicado encontrado.

Cuando 4D abre un archivo de datos, si es necesario construir (o reconstruir) un índice, los duplicados se detectan automáticamente en cualquier campo asociado que se declare único. En este caso, se muestra un cuadro de diálogo de alerta específico antes de abrir la base de datos, ofreciendo al usuario la información necesaria para identificar y eliminar los duplicados:



## Reconstrucción de índices

La conversión de una base 4Dv16 en v17 no necesita de la reconstrucción de los índices (con la excepción de una versión japonesa de 4D).

Pero si la actualización se hace de una versión más antigua (especialmente si actualiza de la librería Unicode (ICU - International Components for Unicode), todos los índices de los campos de tipo texto y de palabras claves de 4D fueron reconstruidos. Esta operación se realiza de forma automática cuando se abre la base por primera vez (advertencia: esta operación puede tomar una cantidad significativa de tiempo).

**Nota:** desde 4D v16, hemos optimizado significativamente el algoritmo de reindexación global de la base de datos. Todos sus procesos fueron revisados y esta operación ahora puede ser hasta dos veces más rápida. La reindexación global es necesaria, por ejemplo, después de reparar la base de datos o cuando se ha eliminado el archivo .4dindx.

## Nota de compatibilidad

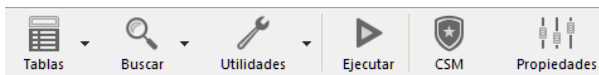
---

Es posible abrir una base de datos en la versión 16.Rx con una versión 4D v16.x y viceversa. Sin embargo, el código que usa las nuevas funcionalidades no funcionará y debería estar deshabilitado. Pero una base de datos convertida a v17 ya no se puede abrir en 4D v16.

**Nota:** similarmente, una base v17 Rx también puede reabrirse con una versión 4D v17.x.

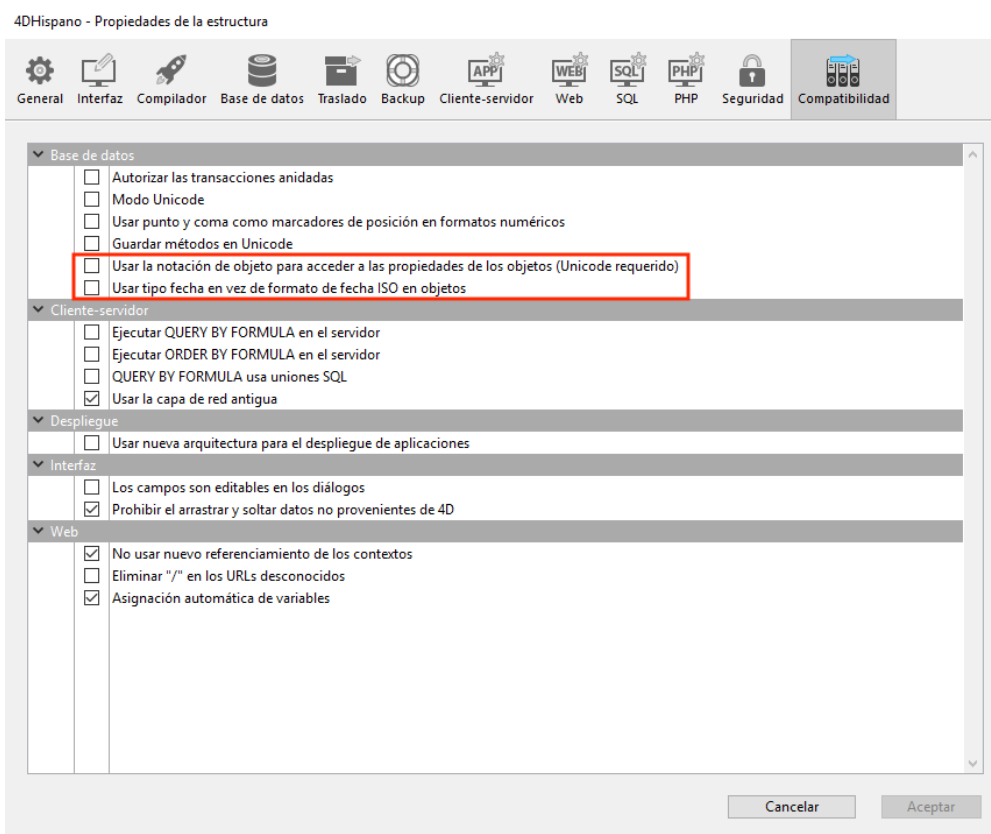
# Diálogo Compatibilidad

Para ir a este diálogo, sólo tiene que hacer clic en el icono "Configuración" en la barra de herramientas principal:



Luego, en la pestaña "Compatibilidad".

Dos nuevas opciones se añadieron en v17:



## Utilizar la notación objeto para acceder a las propiedades de los objetos

Si desea aprovechar la notación de objeto (ver la sección [Uso de la notación objeto](#)), esta opción permite el uso de los caracteres ".", "[" y "]" como símbolos de notación de objeto, para acceder a las propiedades tokenizadas: en los nombres de **variables**, **tablas**, **campos** y **métodos proyecto**.

Activar esta opción es obligatoria en una base de datos creada en una versión anterior a 4D v17 ya que los caracteres ".", "[" y "]" se permitían anteriormente en los nombres de estos elementos y ahora se desaconsejan.

### Atención:

- esta opción requiere que la base de datos esté en Unicode;



- Cuando active esta opción, se recomienda verificar la conformidad de su código existente vía el CSM (ver **Verificar la aplicación**). Para ayudar a los desarrolladores a poner sus aplicaciones conforme con esta regla, la acción **Verificar la aplicación** busca automáticamente la presencia de estos caracteres no deseados en los nombres de variable, tabla, campo y método. Si se detectan estos caracteres, el CSM informa de los errores:

Verificación de nombres de tablas y campos  
Warning: Nombre del método "Tabla.1" contiene puntos o corchetes [](0;0)

En este caso, debe cambiar el nombre de estos elementos en su aplicación.

## Utilizar el tipo fecha en lugar del formato fecha ISO en los objetos

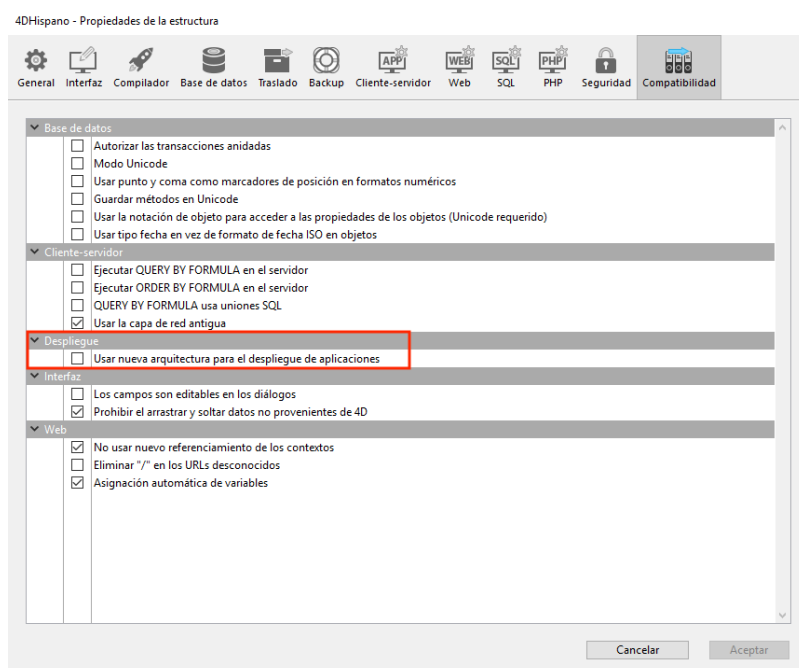
En un objeto, las fechas ahora se pueden definir como fechas y no como cadenas, haciendo que el uso de fechas en objetos sea más fácil e intuitivo. Esto permite utilizar los comandos **OB SET** y **OB Get** sin la constante `Is date`, y se les puede aplicar la notación de objeto como cualquier otro objeto. Para utilizar las fechas en los objetos en una base convertida, simplemente marque la opción "Utilizar el tipo fecha en lugar del formato de fecha ISO en los objetos". Ver [Página Compatibilidad](#).

## Opción eliminada

La opción "**Utilizar los comentarios 4DVAR en lugar de paréntesis**" fue eliminada del diálogo **Compatibilidad** desde 4D v16.

## Opciones antiguas de compatibilidad

Mientras más antigua sea su base de 4D, más opciones se mostrarán:



Para más información vea el tema **[Página Compatibilidad](#)**.

La página Compatibilidad agrupa todos los parámetros relacionados con el mantenimiento de la compatibilidad con versiones anteriores de 4D. Recuerde que el número de opciones mostradas dependerá de la versión de 4D con la cual se creó la base de datos originalmente (2004.x, v11, v12).

**Nota:** esta página no aparece en bases creadas con una versión actual de 4D (bases no convertidas).

- **Los campos no son editables en los diálogos:** en versiones anteriores de 4D, no era posible introducir valores utilizando campos en cajas de diálogo (mostradas, por ejemplo, utilizando el comando DIALOG). Esta limitación se eliminó desde 4D 2004.  
Puede conservar el comportamiento anterior, especialmente si su base de datos utiliza campos en diálogos para mostrar datos. Por defecto, esta opción está seleccionada para las versiones antiguas convertidas a versión 2004 y no está seleccionada para las bases de datos creadas a partir de la versión 2004.
- **Botones de opción agrupados por nombre:** en las versiones anteriores de 4D, el funcionamiento coordinado de un grupo de botones de opción se obtenía dando una primera letra idéntica a las variables asociadas con botones (por ejemplo, *m\_boton1*, *m\_boton2*, *m\_boton3*, etc.). A partir de 4D 2004 se cambió así: para funcionar de manera coordinada, un conjunto de botones de opción simplemente debe estar agrupado en el editor de formularios. Para mayor información, consulte **Botones radio y botones radio imagen**. Este nuevo modo es válido para botones radio, botones radio 3D y .botones radio imagen Por razones de compatibilidad, el modo anterior se conserva por defecto en bases de datos convertidas. Sin embargo, puede forzar el uso del nuevo modo deseleccionando esta opción. Esta opción sólo aparece en bases convertidas de versiones anteriores. Esta seleccionada por defecto. Las bases de datos creadas a partir de la versión 2004 utilizan el nuevo modo.
- **Recargar el formulario para cada registro durante PRINT SELECTION:** en versiones anteriores de 4D, el formulario utilizado durante una impresión utilizando el comando **PRINT SELECTION** se recargaba para cada registro. Esto permitía reinicializar automáticamente todos los parámetros de los objetos que el desarrollador hubiese podido modificar con el lenguaje en el evento de formulario On printing\_detail.  
Para optimizar el rendimiento, este mecanismo se eliminó a partir de 4D 2004. El desarrollador 4D ahora debe reinicializar él mismo los parámetros que quiere en el método de formulario, este funcionamiento es idéntico al de los formularios listados con el evento On display\_detail. No obstante, puede conservar el mecanismo anterior utilizando esta opción. Las bases de datos creadas en versión 2004 utilizan el nuevo modo.
- **No utilizar el nuevo modo de referenciar los contextos:** cuando esta opción no está seleccionada (valor por defecto), el servidor web 4D pone el número del contexto en el URL de base de los documentos HTML enviados. Con el sistema anterior (opción seleccionada), el servidor web 4D envía al navegador el número del contexto por cada elemento de una página, haciendo

menos rápidos los procesos. No obstante esta opción puede seleccionarse por razones de compatibilidad. Recuerde que debe reiniciar la base de datos después de modificar esta opción para que el nuevo funcionamiento tome efecto.

- **Borrar el "/" en las URLs desconocidas:** en las versiones anteriores de 4D, las URLs desconocidas (URLs que no corresponden a páginas existentes ni a URLs especiales) se devolvían en los métodos base On Web Authentication y On Web Connection (\$1) y no comenzaban con el carácter "/". Esta particularidad se eliminó en 4D 2004. Sin embargo, si implementó algoritmos basados en este caso especial y quiere mantenerlos, deseccione esta opción.
- **Prohibir soltar datos que no provengan de 4D:** a partir de la v11, 4D permite arrastrar y soltar selecciones de objetos y/o archivos externos a 4D, como por ejemplo los archivos imagen, en el modo Aplicación. Esta posibilidad debe estar soportada por el código de la base. En las bases de datos convertidas de versiones anteriores de 4D, esta posibilidad puede traer problemas de funcionamiento si el código existente no está adaptado como corresponde. Esta opción permite anticiparse a estos posibles problemas de funcionamiento. Cuando esta opción está seleccionada, se rechaza el soltar objetos externos en formularios 4D. Note que insertar objetos externos aún es posible en los objetos que tienen la opción Soltar automático, cuando la aplicación puede interpretar automáticamente los datos soltados (texto o imagen). Para mayor información, consulte [Arrastrar y soltar](#).
- **Ejecutar QUERY BY FORMULA en el servidor y Ejecutar ORDER BY FORMULA en el servidor:** a partir de 4D v11, por razones de optimización, los comandos de búsquedas y de ordenación "por fórmula" se ejecutan en el servidor; sólo se devuelve el resultado al equipo cliente. Esto concierne a los siguientes comandos: **QUERY BY FORMULA**, **QUERY SELECTION BY FORMULA** y **ORDER BY FORMULA**. En caso de llamada directa de variables en la fórmula, la búsqueda se calcula con el valor de la variable en el equipo cliente. Por ejemplo,

```
QUERY BY FORMULA([tabla];[tabla]campo=lavariabla)
```

se ejecutará en el servidor pero con el contenido de la variable *lavariabla* del cliente. Por otra parte, este principio no aplica a la fórmulas utilizando métodos que, ellos mismos, llaman variables: en este caso el valor de las variables se evalúa en el servidor.

En las bases de datos convertidas, este funcionamiento puede afectar los algoritmos existentes. Por lo tanto, por defecto en este contexto, estos comandos continúan siendo ejecutados en la máquina cliente. Si quiere tomar ventaja del nuevo algoritmo v11 en una base convertida, puede simplemente seleccionar estas opciones.

**Nota:** esta opción puede definirse utilizando el comando **SET DATABASE PARAMETER**.

- **QUERY BY FORMULA usa uniones SQL:** a partir de 4D v11, los comandos **QUERY BY FORMULA** y **QUERY SELECTION BY FORMULA** efectúan uniones basados en el modelo de SQL. Esto significa que no es necesario que exista una relación estructural entre la tabla A y la tabla B para utilizar una fórmula que contenga [Tabla\_A ]campo\_X=[Tabla\_B]campo\_Y. Como este mecanismo puede generar disfunciones en las aplicaciones existentes, está desactivado por defecto en las bases de datos convertidas. Se recomienda activarlo (después de revisar el código de la base) seleccionando esta opción con el fin de beneficiarse de la optimización de los comandos de

búsqueda por fórmula.

**Notas:**

- Cuando el modo "Uniones SQL" está activo, los comandos QUERY BY FORMULA y QUERY SELECTION BY FORMULA utilizan sin embargo las relaciones automáticas definidas en el editor de estructura en los siguientes casos:
  - Si la fórmula no puede descomponerse en elementos de la forma {campo;comparador ;valor}
  - Si se comparan dos campos de la misma tabla.
- Esta opción también puede definirse por proceso utilizando el comando **SET DATABASE PARAMETER**.

- **Autorizar las transacciones anidadas:** activa el soporte de las transacciones multinivel. A partir de la v11, 4D acepta las transacciones anidadas en un número de niveles ilimitado. Como esta nueva operación puede generar problemas en las bases desarrolladas con versiones anteriores de 4D, está desactivado por defecto en las bases convertidas (las transacciones permanecen limitadas a un solo nivel). Si quiere tomar ventaja de las transacciones en varios niveles en una base convertida, debe seleccionar esta opción.

Por defecto, esta opción no está seleccionada. Se especifica para cada base de datos.

**Nota:** esta opción no tiene efecto en las transacciones efectuadas en el motor SQL de 4D. Las transacciones SQL siempre son multinivel.

- **Modo Unicode:** permite activar o desactivar el modo Unicode para la base actual. En modo Unicode, el motor de la base de datos, el lenguaje y los menús manejan nativamente las cadenas de caracteres en Unicode. En modo no Unicode (modo compatibilidad), se utiliza el conjunto de caracteres ASCII. Esta opción permite preservar la compatibilidad de las bases convertidas. Está seleccionada por defecto para las bases creadas con 4D v11 y superiores y no está seleccionada en las bases convertidas.

**Notas:**

- Esta opción es específica para cada base. Por lo tanto es posible hacer cohabitar una base Unicode con los componentes no Unicode (o viceversa) en modo interpretado.
- También es posible configurar el modo Unicode utilizando los comandos **SET DATABASE PARAMETER**.
- Las versiones 64 bits de 4D no soportan modo de compatibilidad ASCII.

Las características específicas del soporte Unicode en 4D se detallan en el manual *Lenguaje*. Para mayor información consulte **EXPORT TEXT**.

- **Utilice los parámetros del sistema en formatos numéricos:** a partir de v11, 4D utiliza los parámetros sistema regionales para los formatos de visualización numéricos (ver "Formatos de campos numéricos" en **Formatos de salida**). 4D reemplaza automáticamente los caracteres ",", "y" y "." en los formatos de visualización numérica por el separador de miles y el separador decimal definido en el sistema operativo, respectivamente. El punto y la coma son entonces considerados como los caracteres de marcadores de posición, siguiendo el ejemplo de 0 o #. En las versiones anteriores de 4D, los formatos de visualización numéricos no tienen en cuenta los parámetros regionales del sistema. Por ejemplo, el formato "###,##0.00" es un formato válido para un sistema americano. Sin embargo, cuando se aplica a un valor numérico mostrado en un sistema francés o suizo, el resultado es incorrecto. En las bases de datos convertidas, por compatibilidad, este nuevo mecanismo no está activado. Para beneficiarse de él, debe seleccionar esta opción.
- **Asignación automática de variables:** en versiones anteriores de 4D, un mecanismo estándar del servidor web permite volver a copiar automáticamente

en las variables proceso 4D el valor de las variables enviadas por medio de un formulario HTTP o un URL de tipo GET. En modo interpretado, el valor de toda variable recibida se copia directamente en una variable proceso 4D con el mismo nombre, en modo compilado, las variables deben haber sido previamente declaradas en un método proyecto *COMPILER\_WEB*.

A partir de 4D v13.4, este mecanismo es obsoleto y ya no está disponible en las nuevas bases de datos. Por razones de compatibilidad, se mantiene en las bases convertidas, pero puede desactivarlo desmarcando esta opción de compatibilidad. Ahora se recomienda utilizar los comandos dedicados **WEB GET VARIABLES** o **WEB GET BODY PART**.

- **Use legacy network layer:** Starting with release v15, 4D applications propose a new network layer, named *ServerNet*, to handle communications between 4D Server and remote 4D machines (clients). The former network layer has become obsolete, but it is kept to ensure compatibility with existing databases. Using this option, you can enable or disable the former network layer at any time in your converted 4D Server applications depending on your needs, for example, when migrating your client applications (see the **Opciones red y cliente-servidor** section). *ServerNet* is used automatically for new databases, and disabled by default in converted databases (option checked).

Note that in case of a modification, you need to restart the application for the change to be taken into account. Any client applications that were logged must also be restarted to be able to connect with the new network layer (the minimum client version for using the *ServerNet* layer is 4D v14 R4, see the **Opciones red y cliente-servidor** section).

**Note:** This option can also be managed by programming using the **SET DATABASE PARAMETER** command.

- **Utilizar la capa de red antigua:** a partir del lanzamiento de v15, las aplicaciones 4D contienen una nueva capa de red, llamada *ServerNet*, para manejar las comunicaciones entre 4D Server y las máquinas 4D remotas (clientes). La capa de red antigua ha quedado obsoleta, pero se conserva para asegurar la compatibilidad con las bases de datos existentes. Mediante esta opción, puede activar o desactivar la capa de red antigua en cualquier momento en las aplicaciones 4D Server convertidas en función de sus necesidades, por ejemplo, al migrar las aplicaciones cliente (ver la sección **Opciones red y cliente-servidor**). *ServerNet* se utiliza automáticamente para las nuevas bases y se desactiva por defecto en las bases convertidas (opción seleccionada).

Tenga en cuenta que en caso de una modificación, es necesario reiniciar la aplicación para que el cambio se tenga en cuenta. Todas las aplicaciones cliente que se han conectado también se deben reiniciar para poder conectar con la nueva capa de red (la versión de cliente mínima para usar la capa *ServerNet* es 4D v14 R4, consulte la sección **Opciones red y cliente-servidor**).

**Nota:** esta opción también se puede administrar por programación utilizando el comando **SET DATABASE PARAMETER**.

- **Guardar los métodos como Unicode:** le permite guardar cadenas de métodos 4D en Unicode. En las versiones de 4D anteriores a la v15, las cadenas presentes en el código de los métodos (fórmulas, nombres de variables y de métodos, comentarios, etc.) se guardaban utilizando la codificación local actual. Esta codificación podría causar problemas, sobre todo cuando se comparte código 4D entre desarrolladores de diferentes países: por ejemplo, si un desarrollador francés escribió código 4D con acentos y luego envía la base a un desarrollador inglés, se perderían los acentos. Serios problemas también podrían ocurrir con código escrito en versiones en japonés. El guardar los métodos en Unicode resuelve todos estos problemas y hace posible el intercambio de código 4D con caracteres locales específicos. Le recomendamos

activar la opción de modo Unicode para métodos tan pronto como sea posible en sus bases existentes, especialmente si trabaja en un entorno internacional.

**Notas:**

- Esta funcionalidad se aplica al lenguaje y su interpretación. Algunas ventanas del editor 4D, tales como Lista de propiedades, aún utilizan codificación local actual y por lo tanto pueden mostrar ciertas cadenas incorrectamente. Sin embargo, esto no afecta a la ejecución del código.
- Puede activar o desactivar esta opción en cualquier momento; solo los métodos guardados posteriormente se ven afectados.
- Las versiones 64 bits 4D no soportan modo de compatibilidad ASCII.
- **Utilizar el tipo fecha en lugar del formato de fecha ISO en objetos:** le permite almacenar las fechas en los atributos objeto como tipo fecha en lugar de texto en formato ISO. En las versiones anteriores de 4D, las fechas en atributos objeto solo podían almacenarse como cadenas. A partir de 4D v16 R6, las fechas en los atributos objeto se pueden almacenar en tipo fecha. Puede habilitar este comportamiento marcando esta opción (las fechas almacenadas previamente en el formato ISO permanecerán "tal cual", pero las nuevas fechas guardadas estarán en el tipo fecha). Esta función también se puede administrar por programación, por proceso, utilizando el comando **SET DATABASE PARAMETER** y los selectores Date type, [`#cst id="3546829"/]` y String type without time zone.
- **Utilizar la nueva arquitectura para las aplicaciones desplegadas:** esta opción está disponible para todas las aplicaciones a partir de 4D v15 R4. Activa o desactiva nuevos mecanismos relacionados con el despliegue de las aplicaciones 4D (debe ser definido en la máquina que genera la aplicación final). Los mecanismos controlados por esta opción se describen en las secciones **Último archivo de datos abierto** y **Gestión de la conexión de las aplicaciones clientes**. Esta opción está desactivada de forma predeterminada en las aplicaciones convertidas. Para beneficiarse de estos nuevos mecanismos, es necesario seleccionarlos de forma explícita.



# Cambios de comportamiento

## Object Get action

---

Los comandos **OBJECT Get action** y **OBJECT SET ACTION** ahora utilizan constantes con valores de tipo cadena en lugar de entero largo (cuando se usa en objetos de formulario).

Código anterior:

```
ARRAY TEXT($arrObjects;0)
FORM GET OBJECTS($arrObjects)
For($i;1;Size of array($arrObjects))
    If(OBJECT Get action(*;$arrObjects{$i})=Object No standard action)
        OBJECT SET ACTION(*;$arrObjects{$i};Object Cancel action)
    End if
End for
```

Código después de la conversión a v17:

```
ARRAY TEXT($arrObjects;0)
FORM GET OBJECTS($arrObjects)
For($i;1;Size of array($arrObjects))
    If(_o_OBJECT Get action(*;$arrObjects{$i})=_o_Object No standard action)
        OBJECT SET ACTION(*;$arrObjects{$i};_o_Object Cancel action)
    End if
End for
```

**Nota:** solo la función \_o\_Object Get action ha sido declarada obsoleta porque en el caso de la compilación, es necesario establecer el tipo de resultado de antemano. Esto se hizo para que pueda considerar cambiar su código (ver a continuación), de lo contrario la compilación sería imposible.

Nuevo código: comando y constantes obsoletas a remplazar por:

```
ARRAY TEXT($arrObjects;0)
FORM GET OBJECTS($arrObjects)
For($i;1;Size of array($arrObjects))
    If(OBJECT Get action(*;$arrObjects{$i})=ak none)
        OBJECT SET ACTION(*;$arrObjects{$i};ak cancel)
    End if
End for
```



## Horas expresadas en segundos

---

A partir de v17, las horas 4D (tipo **C\_TIME**) manipuladas a través de las propiedades del objeto se convierten en **número de segundos**. En versiones anteriores, se convertían a milisegundos.

Este cambio se aplica a todas las horas convertidas a/desde objetos o colecciones utilizando notación de objetos, comandos como **OB SET** y **OB Get**, **QUERY BY ATTRIBUTE**, o los comandos JSON como **JSON Stringify** y **JSON Parse**. También impacta a las conversiones horas/digitales en las siguientes funcionalidades 4D:

- Áreas web (vía JavaScript),
- 4D Mobile,
- SQL (función **CAST**)

Para facilitar la migración de las bases de datos 4D (especialmente si se han almacenado horas en los atributos del campo objeto), se puede usar una nueva opción de compatibilidad para restaurar el comportamiento anterior de la sesión:

```
//restaurar el comportamiento anterior
//que se colocará en el método base On opening / On server startup
SET DATABASE PARAMETER(Times inside objects;Times in milliseconds)
```

## Modificaciones XML

---

- **Resolución de las entidades externas**

A partir de 4D v16 R3, para mejorar la seguridad, la resolución de entidades externas no está activada por defecto en los documentos XML (la declaración de una entidad externa genera un error de análisis). Para restablecer el comportamiento anterior, utilice el nuevo selector XML external entity resolution con el valor XML enabled para el comando **XML SET OPTIONS**.

- **Sensibilidad a mayúsculas y minúsculas**

A partir de 4D v16 R4, los comandos **DOM Get XML element** y **DOM Count XML elements** distinguen entre mayúsculas y minúsculas por defecto. Un nuevo selector le permite volver al comportamiento anterior: XML DOM case sensitivity con los siguientes valores disponibles:

- XML case sensitive (valor por defecto): los comandos son sensibles a las mayúsculas y minúsculas
- XML case insensitive: los comandos no son sensibles a mayúsculas y minúsculas

## 4D Write Pro y atributos imagen

---

Los atributos wk image, wk list style image y wk background image utilizados con los comandos **WP GET ATTRIBUTES** o **OB Get** devuelven la imagen en sí y no su URL cuando la imagen no está declarada.

Para buscar la URL de la imagen, utilice los nuevos atributos wk image url, wk list style image url y wk background image url.



# Cambios de nombre o de tema

## 4D Write Pro

---

- Los párrafos WP Get se renombraron **WP Create paragraph range**.
- Las imágenes WP Get fueron renombradas **WP Create picture range**.

## Constantes acciones estándar

---

Las antiguas constantes numéricas del tema "**Acción estándar**" son obsoletas (con el prefijo "\_o\_"). A partir de 4D v16 R3, se recomienda utilizar las constantes de tipo texto, con el prefijo "ak ...".

# **Funcionalidades obsoletas**

## **Comandos obsoletos**

---

Para encontrar todos los comandos obsoletos en sus bases, puede buscar "\_o\_" en **Buscar en diseño** en el menú **Edición** o el área **Buscar en diseño** de la barra de herramientas. Le recomendamos especialmente que reemplace cualquier comando obsoleto con nuevos comandos o funciones. Sin embargo, tenga en cuenta que los comandos que no están indicados como "desactivados" seguirán funcionando de forma temporal.

Para una lista de todos los comandos obsoletos: [Lenguaje: comandos obsoletos o eliminados](#)

## **Comandos obsoletos en 4D v17**

---

- **Comandos 4D**

Nombre anterior	Nuevo nombre	Comentarios	Obsoleto desde	Estado actual
<b>Object Get action</b>	<b>OBJECT Get action</b> (o <b>_o_OBJECT Get action</b> en caso de conversión), con cambio de parámetro (cambió de un parámetro con valor de tipo entero largo a un parámetro con valor de tipo cadena)	Al convertirlo, el comando <b>_o_Object read action</b> indica un cambio de comportamiento (ver el capítulo "Cambio de comportamiento").	v16R3	Obsoleto
<b>PICTURE TO GIF</b>	<b>_o_PICTURE TO GIF</b>	<b>PICTURE TO BLOB</b>	v16R5	Obsoleto
<b>Type document</b>	<b>_o_Document type</b>	utilice el comando <b>Path to object</b>	v16R6	Obsoleto
<b>Document creator</b>	<b>_o_Document creator</b>	utilice el comando <b>Path to object</b>	v16R6	Obsoleto
<b>SET DOCUMENT TYPE</b>	<b>_o_Document type</b>	utilice el comando <b>Object to path</b>	v16R6	Obsoleto
<b>SET DOCUMENT CREATOR</b>	<b>_o_SET DOCUMENT CREATOR</b>	utilice el comando <b>Object to path</b>	v16R6	Obsoleto
<b>MAP FILE TYPES</b>	<b>_o_MAP FILE TYPES</b>	Utilice UTIs e info.plist	v16R6	Obsoleto
<b>Gestalt</b>	<b>_o_Gestalt</b>	fue reemplazado por <b>Get system info / Is macOS / Is Windows</b>	v17	Obsoleto
<b>PLATFORM PROPERTIES</b>	<b>_o_PLATFORM PROPERTIES</b>	fue reemplazado por <b>Get system info / Is macOS / Is Windows</b>	v17	Obsoleto

- **Constantes 4D**

Nombre anterior	Nuevo nombre	Comentarios	Obsoleto desde	Estado actual
<u>lk</u> <u>affichage</u> <u>barre déf</u> <u>hor</u>	<u>_o lk</u> <u>display.</u> <u>hor</u> <u>scrollbar</u>	<b>OBJECT GET SCROLLBAR</b>	v16R3	Obsoleto
<u>lk</u> <u>affichage</u> <u>barre déf</u> <u>ver</u>	<u>_o lk</u> <u>display.</u> <u>ver</u> <u>scrollbar</u>	<b>OBJECT GET SCROLLBAR</b>	v16R3	Obsoleto
<u>lk</u> <u>position</u> <u>barre déf</u> <u>hor</u>	<u>_o lk</u> <u>hor</u> <u>scrollbar</u> <u>position</u>	<b>OBJECT GET SCROLL POSITION</b>	v16R3	Obsoleto
<u>lk</u> <u>position</u> <u>barre déf</u> <u>ver</u>	<u>_o lk</u> <u>ver</u> <u>scrollbar</u> <u>position</u>	<b>OBJECT GET SCROLL POSITION</b>	v16R3	Obsoleto
<u>lk</u> <u>hauteur</u> <u> pied</u>	<u>_o lk</u> <u>footer</u> <u>height</u>	<b>LISTBOX Get footers height</b>	v16R3	Obsoleto
<u>lk</u> <u>hauteur</u> <u>entête</u>	<u>_o lk</u> <u>header</u> <u>height</u>	<b>LISTBOX Get headers height</b>	v16R3	Obsoleto
<u>_o Objet</u> <u>xxx</u>	<u>ak xxx</u>	Constantes que soportan acciones estándar. Ver, especialmente los comandos <b>OBJECT Get action</b> y <b>OBJECT SET ACTION</b> .	v16R3	Obsoleto

#### • 4D Internet Commands

Nombre anterior	Nuevo nombre	Comentarios	Obsoleto desde	Estado actual
<i>FTP_Progress</i>	devuelve un error si se llama	El parámetro progress ya no es soportado por los comandos FTP_Append, FTP_Receive, FTP_Send.	v16R3	Desactivado

## **Funcionalidades desactivadas**

---

- El plug-in **4D Pack** se ha eliminado a partir de 4D v16R2.  
**Nota:** 4D Pack de la versión 16.x es compatible con 4D v16 Rx.
- **FTP\_Progress:** en 4D Internet Commands, este comando se ha desactivado a partir de 4D v16R2. Ahora devuelve un error -2201 si se llama.  
**Nota:** el parámetro *progress* ya no se soporta para los siguientes comandos:  
FTP\_Append  
FTP\_Receive  
FTP\_Send

# Cambiar de 32 bits a 64 bits

---

4D v17 funciona en 32 bits o 64-bits, macOS o Windows.

Actualizar una aplicación 4D 32 bits a una versión 64 bits requiere un trabajo de preparación.

Toda la línea de productos 4D 64 bits ya no depende de la librería *Mac2Win* de Altura. **Mac2Win de Altura ahora está completamente eliminado de las versiones 64 bits de 4D Developer Edition y 4D Volume Desktop, lo que permite que estos productos utilicen completamente las API Windows modernas:**

- 4D Developer Edition y 4D Volume Desktop Windows 64 bits en v17 (desde v16R2)
- 4D Server Windows 64 bits en 4D v17 (desde v16 R4)

**Nota:** las versiones Windows 32 bits de 4D aún utilizan Mac2Win.

4D Developer Edition 64 bits integra los nuevos editores de etiquetas, de informes rápidos y de importación y exportación, modernos, intuitivos y fácil de usar. Puede lanzar varias instancias de 4D Developer Edition 64 bits en su ordenador sin tener que instalar la aplicación dos veces.

Si su aplicación funciona con un 4D Server 64 bits Windows o macOS, la mayor parte del trabajo ya está hecho. Las aplicaciones monopuesto 64 bits pueden requerir algunos pasos adicionales. Esta sección ofrece una lista de comprobación paso a paso para ayudarle a verificar todos los puntos necesarios antes y después de la actualización.

Varias funcionalidades se han actualizado, desactivado o incluso se han declarado obsoletas con la migración a 64 bits de nuestros productos. Todos los detalles se listan en el párrafo [[#title id="8807" anchor="3080192"/\]](#) de la sección **Utilizar 4D Developer Edition 64 bits**.

**Nota:** al igual que con cualquier proceso de actualización, es una buena práctica utilizar el CSM y lanzar un proceso de verificación antes de cada paso importante para asegurarse de que tanto los datos como la estructura son válidos.

## Verificar los plug-ins

---

El primer requisito consiste en actualizar sus plug-ins (si los hay) a su versión 64 bits:

- **Plug-ins 4D:**

Todos los plug-ins ya existen en versiones 64 bits, excepto 4D Write y 4D View.

- Si su aplicación utiliza 4D Write, debe considerar migrar su código a 4D Write Pro. Una buena práctica es guardar su código 32 bits existente e iniciar un nuevo módulo basado en 4D Write Pro 64 bits. Si, durante un periodo transicional, desea tener a la vez sus documentos 4D Write y 4D Write Pro, debe utilizar 4D v17 32 bits.



- Si su aplicación utiliza 4D View, deberá utilizar las funcionalidades de 4D View Pro u otras alternativas.
- **Plug-ins de terceros:**  
Contacte a sus proveedores para obtener las versiones 64 bits.

## Preparar la actualización a la versión 32 bits

---

1. Actualice su aplicación a la última versión 32 bits, por ejemplo, 4D v17 32 bits.
2. Asegúrese de que el modo Unicode esté activo.
3. Convierta toda imagen PICT/cicn/QuickTime. Para detectar los formatos de imagen obsoletos en sus datos, puede utilizar el comando **GET PICTURE FORMATS**. También es necesario reemplazar todos los formatos de imagen no soportados en la estructura de su base. Una verificación con el CSM detectará imágenes obsoletas en archivos de recursos para los botones de imagen y 3D, así como también para imágenes estáticas.
4. Reemplace las funcionalidades basadas en XSLT (comandos **\_o\_XSLT APPLY TRANSFORMATION**, **\_o\_XSLT SET PARAMETER** o **\_o\_XSLT GET ERROR**), por el comando **PROCESS 4D TAGS** por ejemplo.
5. Reemplace las llamadas **\_o\_Font number** por llamadas de nombre de fuentes.
6. Elimine todo código que cree o modifique archivos de recursos.

En este punto, usted está listo para abrir su base de datos con una versión 64 bits de 4D.

## Abrir y verificar la base en versión 64 bits

---

1. Abra su aplicación con una versión 64 bits de 4D Developer Edition.
2. Si utiliza el motor de renderización WebKit integrado para sus áreas Web, verifique que pase automáticamente al motor de renderización del sistema (el acceso a los métodos 4D con \$4d sigue siendo válido).
3. Si su código utiliza la Mac spool file format option del comando **SET PRINT OPTION**, debe reemplazarlo por una llamada a **SET CURRENT PRINTER** con la constante Generic PDF driver.
4. Verifique las llamadas y usos del editor de etiquetas (consulte **Editor de etiquetas (64 bits)**).
5. Verifique las llamadas y usos del editor de informes rápidos (consulte **Informes rápidos (64 bits)**).

Su aplicación es totalmente compatible con 64 bits y puede beneficiarse de todas las nuevas funcionalidades de 4D 64 bits.

## Beneficios de las funcionalidades 64 bits

---

En particular:

- La **arquitectura 64 bits** empuja los límites de la memoria caché de la base de datos. Mejora el rendimiento de su base simplemente utilizando una caché más grande.  
Adopta **poderosas funcionalidades** 64 bits, como procesos apropiativos, objetos de formulario animados o nuevas funcionalidades de impresión.  
Construya sus aplicaciones con 4D Runtime Volume License 64 bits.  
Utilice 4D Server 64 bits Win y macOS en versión final, consulte las secciones

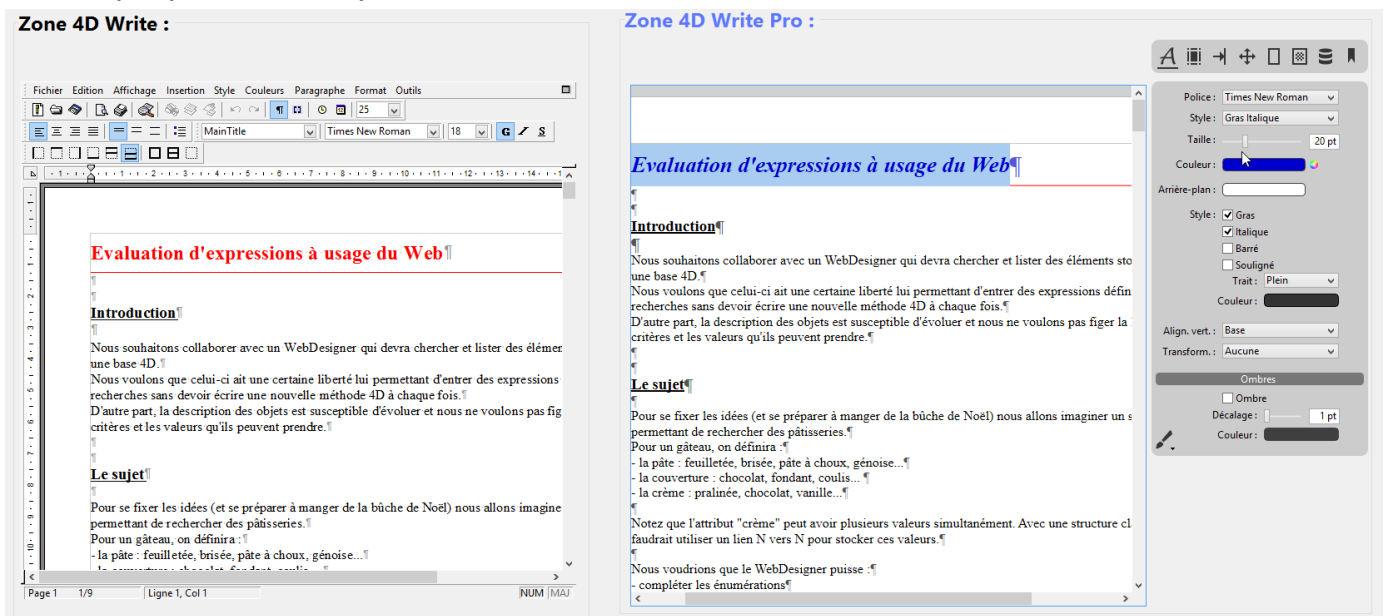
## Utilizar 4D Server 64 bits para OS X y Utilizar 4D Server 64 bits (Windows)

- **Nuevo editor de informes rápidos**, compatible con los informes creados con versiones anteriores. Ver **Editor de etiquetas (64 bits)**.
- **Nuevo editor de etiquetas**, compatible con los archivos de etiquetas creados con versiones anteriores. Ver **Informes rápidos (64 bits)**.
- Crea **gráficos** utilizando un parámetro de tipo Objeto con el comando **GRAPH**.

# Convertir los documentos 4D Write en 4D Write Pro

## Convertir un documento 4D Write

4D Write Pro puede abrir y convertir documentos 4D Write conservando la mayoría de sus propiedades específicas:



En la imagen de arriba, tenemos un área 4D Write a la izquierda y un área de 4D Write Pro a la derecha (creada con el nuevo objeto de la librería de objetos - ver abajo). El contenido del área 4D Write se recuperó simplemente usando el comando **WP New**:

```
//recuperamos el contenido del área 4D Write en el área 4D Write Pro  
[WRITEAREAS]AreaNTWP:=WP New([WRITEAREAS]AreaNT_)
```

Pero como 4D Write sólo se puede utilizar con versiones 4D 32 bits, debe convertir sus documentos 4D Write antes de cambiar a la versión 64 bits.

A diferencia de 4D Write, 4D Write Pro no es un plug-in, sino que está totalmente integrado en 4D. Tenga en cuenta que 4D Write Pro utiliza la misma licencia que 4D Write. Necesita tener esta licencia instalada en su aplicación para habilitar la funcionalidad.

Los objetos 4D Write Pro permiten recuperar los documentos 4D Write de dos maneras:

- Para los archivos 4D Write almacenados en disco, puede utilizar el comando **WP Import document**. Los archivos 4D Write (.4w6, .4w7 y plantilla .4wt) deben convertirse en archivo 4D Write Pro (.4wp).

```
// Primero, convertir archivos .4w6 en archivos .4w7 con comandos 4D Write
$offscreen:=WR New offscreen area
WR OPEN DOCUMENT($offscreen;"myFile.4w6";"4WR6")
WR SAVE DOCUMENT($offscreen;"myFile.4w7";"4WR7")
WR DELETE OFFSCREEN AREA($offscreen)
```

```
// Luego convertir .4w7 en .4wp con comandos 4D Write Pro
C_OBJECT($docWritePro)
$docWritePro:=WP Import document("myFile.4w7")
WP EXPORT DOCUMENT($docWritePro;"myFile.4wp")
```

- Para los archivos 4D Write almacenados en el archivo de datos, puede utilizar el comando [#title id="3680"/]. Los archivos 4D Write almacenados en campos BLOB o Imagen deben transferirse a un campo Objeto.

```
// De un campo Imagen a un campo Objeto, pasando por un BLOB
// [DocWRITE]WritePictArea_ es un campo imagen
// $Blob is a BLOB
// [DocWRITE]WriteProArea es un campo Objeto
$offscreen:=WR New offscreen area
WR PICTURE TO AREA($offscreen;[DocWRITE]WritePictArea_)
$Blob:=WR Area to blob($offscreen;1)
[DocWRITE]WriteProArea :=WP New($Blob)
WR DELETE OFFSCREEN AREA($offscreen)
```

```
// De un campo BLOB a un campo Objeto
// [DocWRITE]WriteBLOBArea_ is a BLOB field
// [DocWRITE]WriteProArea is an object field
[DocWRITE]WriteProArea :=WP New([DocWRITE]WriteBLOBArea_)
```

### Notas de compatibilidad:

- Sólo se soportan los documentos 4D Write de última generación ( "4D Write v7").
- Compruebe las funcionalidades y objetos importables consultando: **¿Qué propiedades se conservan de 4D Write?**
- Copiar y pegar desde un documento 4D Write a un área 4D Write Pro por el momento no se soporta. Un documento 4D Write sólo se puede importar utilizando los comandos del lenguaje 4D Write Pro.
- En Windows, las funcionalidades de 4D Write Pro se apoyan en Direct2D. Con máquinas en Windows 7 o Windows Server 2008, asegúrese de que se ha

instalado el componente *Platform Update for Windows* para que pueda beneficiarse de la versión requerida de Direct2D.

## Filtrado de expresiones 4D

---

El filtrado no estaba activo para los documentos 4D Write Pro en versiones anteriores. Si sus documentos 4D Write Pro hacen referencia a métodos 4D, ya no se evaluarán correctamente una vez que se conviertan a 4D v16 o superior. Los mensajes se mostrarán como "## Error # 48".

En este caso, debe agregar los métodos a la lista de métodos permitidos utilizando el comando **SET ALLOWED METHODS**.

## Comandos modificados

---

Se han añadido nuevos comandos y los ya existentes han evolucionado para trabajar con 4D Write Pro:

- Regla horizontal: para ajustar márgenes, indentaciones y pestañas.
- Barra de herramientas personalizada: extensión del mecanismo de las acciones estándar.
- Actualización del comando **Dynamic pop up menu**: para designar su propio menú contextual 4D Write Pro basado en las acciones estándar.
- Gestión de tablas: **WP Insert table**, **WP Table append row**, **WP Table get rows**, **WP Table get columns**, **WP Table get cells**
- Hipervínculos: nuevo atributo wk link url para los comandos **WP SET ATTRIBUTES** y **WP GET ATTRIBUTES**.
- Inserción de imagen: comando **WP Add picture**, e imagen de fondo en página completa con el comando **WP SET ATTRIBUTES** (atributo: wk paper box).
- Gestión de encabezados y de pie de página: **WP Get header**, **WP Get footer**, **WP Get body**.
- Comandos para desplazar el cursor (**WP SET FRAME**) y ubicarlo (**WP Get frame**).
- Caracteres iniciales para pestañas (con el comando **WP SET ATTRIBUTES**).

# Conversión de documentos 4D View a 4D View Pro

## Conversión

---

4D v16 R6 introdujo el primer paso hacia la conversión de sus documentos 4D View a 4D View Pro **pre-versión**. Gracias al nuevo comando **VP Convert from 4D View**, la mayoría de las propiedades y la información almacenada en los documentos 4D View se convierten automáticamente, incluida la estructura del documento, los valores, los formatos, los estilos, los bordes y las fórmulas.

Estos puntos se detallan en las siguientes páginas del manual de referencia de 4D View Pro:

- **Conversión de documentos 4D View, Proceso de conversión**
- **Detalles de la conversión.**

## Área 4D View Pro

---

Un nuevo objeto de formulario 4D View Pro (**Objeto 4D View Pro**) y los nuevos comandos asociados están disponibles:

- Cree un nuevo documento con **VP NEW DOCUMENT**,
- Guárdelo en el disco con **VP EXPORT DOCUMENT** o en la base de datos con **VP Export to object**,
- Luego vuelva a abrirlo con **VP IMPORT DOCUMENT** o **VP IMPORT FROM OBJECT**.

# Anexo: Notas del lanzamiento 4D v16 Rx

---

Durante el ciclo de las versiones R-release, varias librerías y componentes utilizados por 4D se actualizan y se registran problemas conocidos. Esta información, publicada inicialmente a través de las "Notas de la versión" de cada versión, se resume en esta página.

## 4D v16 R2

---

4D Developer y 4D Volume Desktop para Windows 64 bits están en versión final a partir de 4D v16 R2.

El plug-in 4D Pack ya no se ofrece en los instaladores. Tenga en cuenta que 4D Pack para 4D v16.x es compatible con 4D v16 R.x

## 4D v16 R3

---

La antigua librería Altura se ha eliminado de 4D Developer Edition y 4D Volume Desktop Windows 64 bits.

XML: cambio de la resolución de las entidades externas, nueva opción para [**XML SET OPTIONS**] (cf. **Modificaciones XML** ).

## 4D v16 R4

---

Xerces: actualización a la versión 3.1.4 (librería "Código abierto" utilizada internamente para gestionar el XML).

La anterior librería Altura se ha eliminado de 4D Server Windows 64 bits. A partir de ahora, toda la línea de productos 64 bits está "libre de Altura".

XML: a partir de 4D v16 R4, los comandos **DOM Get XML element** y **DOM Count XML elements** distinguen entre mayúsculas y minúsculas por defecto. El nuevo selector XML DOM case sensitivity para el comando **XML SET OPTIONS** permite volver al comportamiento anterior (ver **Modificaciones XML** ).

## 4D v16 R5

---

CEF: actualización a la versión 301. Chromium Embedded Framework (CEF).

libldap: actualiza a la versión 2.45.

libsasl: actualización a la versión 2.1.17. (Capa de seguridad de autenticación simple)

libzip: actualización a la versión 1.2.

zlib: actualización a la versión 1.2.11. (librería de software de compresión de datos).

Pantalla HDPI para Windows 10 y plug-ins anteriores:

- Los anteriores plug-ins 4D Write y 4D View no están certificados para pantallas High DPI en Windows 10 (1709). Recomendamos utilizar la línea de productos 64 bits en este contexto (4D Write Pro y 4D View Pro).

Área web en 4D 32 bits y macOS 10.13:

- Es posible que el área web ya no esté enfocada al cambiar de fondo a primer plano. Recomendamos utilizar la línea de productos de 64 bits en este contexto.

El dibujo de PDFs en un área web utiliza 4D 32 bits para macOS 10.13:

- Es posible que el área web no muestre un PDF con el plug-in del visor de PDF para macOS 10.13. Recomendamos utilizar la línea de productos 64 bits en este contexto.

## 4D v16 R6

---

Hunspell: Actualización a la versión 1.6.2.

Los atributos wk image, wk list style image y wk background image utilizados con los comandos **WP GET ATTRIBUTES** o **OB Get** devuelven la imagen en sí y no su URL cuando la variable imagen no está declarada. Para devolver la URL de la imagen, utilice los nuevos atributos wk image url, wk list style image url y wk background image url.

WP Get paragraphs ha sido renombrado **WP Create paragraph range**.

WP Get pictures ha sido renombrado **WP Create picture range**.

## 4D v17

---



# Anexo: métodos útiles para la conversión

## Para generar un archivo disco que liste campos únicos no indexados

---

[TechTip](#) para generar un archivo disco que liste campos únicos no indexados:

```
C_LONGINT($maxTableName_1;$currentTable_1)
C_LONGINT($maxFieldCount_1;$currentField_1)
C_LONGINT($dontCare_1) // para los valores GET FIELD PROPERTIES que no se utilizan.
C_BOOLEAN($dontCare_f;$isIndexed_f;$isUnique_f)
C_TEXT($logHeader_t;$logRecord_t;$logfile_t)
C_TEXT($delim_t;$lf_t)
C_TIME($logfile_h)
C_TEXT($tableName_t;$fieldName_t;$note_t)

$delim_t:=Char(Tab)
$lf_t:=Char(Carriage return)+Char(Line feed)

$logHeader_t:="Unique fields without index:"+$lf_t

$logfile_t:=Get 4D folder(Logs folder)+"UniqueNotIndexed.txt"

$logfile_h:=Create document($logfile_t)

If(OK=1)

    SEND PACKET($logfile_h;$logHeader_t)

    $maxTableName_1:=Get last table number

    For($currentTable_1;1;$maxTableName_1)
        If(Is table number valid($currentTable_1))
            $maxFieldCount_1:=Get last field number(Table($currentTable_1))
            For($currentField_1;1;$maxFieldCount_1)
                If(Is field number valid($currentTable_1;$currentField_1))

// Tenga en cuenta los siguientes saltos de línea sobre dos líneas en el texto,
// es una declaración en el método:
                GET FIELD PROPERTIES($currentTable_1;$currentField_1;$dontCare_1;\
                $dontCare_1;$isIndexed_f;$isUnique_f;$dontCare_f)
```

```

If(($isUnique_f) & (Not($isIndexed_f)))

    $tableName_t:=Table name(Table($currentTable_l))
    $fieldName_t:=Field name(Field($currentTable_l;$currentField_l))

    $logRecord_t:="[ "+$tableName_t+" ]"+$fieldName_t+$lf_t

    SEND PACKET($logfile_h;$logRecord_t)

        End if
    End if
End for
End if
End for

CLOSE DOCUMENT($logfile_h)
SHOW ON DISK($logfile_t)
End if

```

## Para crear los índices faltantes

[TechTip](#) para crear índices faltantes en los campos declarados "únicos" pero no indexados:

```

C_LONGINT($maxTableNumber_l;$currentTable_l)
C_LONGINT($maxFieldCount_l;$currentField_l)
C_LONGINT($dontCare_l) // para valores GET FIELD PROPERTIES que no son utilizados.
C_BOOLEAN($dontCare_f;$isIndexed_f;$isUnique_f)
C_TEXT($logHeader_t;$logRecord_t;$logfile_t)
C_TEXT($delim_t;$lf_t)
C_TIME($logfile_h)
C_TEXT($tableName_t;$fieldName_t;$note_t)

```

```
$maxTableNumber_l:=Get last table number
```

```
For($currentTable_l;1;$maxTableNumber_l)
```

```
    If(Is table number valid($currentTable_l))
```

```
        $maxFieldCount_l:=Get last field number(Table($currentTable_l))
```

```
        For($currentField_l;1;$maxFieldCount_l)
```

```
            If(Is field number valid($currentTable_l;$currentField_l))</p><p> // Tenga en
```

cuenta los siguientes saltos de línea sobre dos líneas en el texto,

// es una declaración en el método:

```

                GET FIELD PROPERTIES($currentTable_l;$currentField_l;$dontCare_l;\
                $dontCare_l;$isIndexed_f;$isUnique_f;$dontCare_f)

```

```
If(($isUnique_f) & (Not($isIndexed_f)))
```

```
    $tablePtr:=Table($currentTable_l)
```

```
    $fieldPtr:=Field($currentTable_l;$currentField_l)
```

```
    $tableName_t:=Table name($tablePtr)
```

```
    $fieldName_t:=Field name($fieldPtr)
```

```
    $indexName_t:="[ "+$tableName_t+" ]+$fieldName_t+" indexed for  
uniqueness (kb#77023)"
```

```
    ARRAY POINTER($fieldsArray_p;1)
```

```
    $fieldsArray_p{1}:= $fieldPtr
```

```
    CREATE INDEX($tablePtr->$fieldsArray_p;Standard BTree  
Index;$indexName_t;*)
```

```
End if
```

```
End if
```

```
End for
```

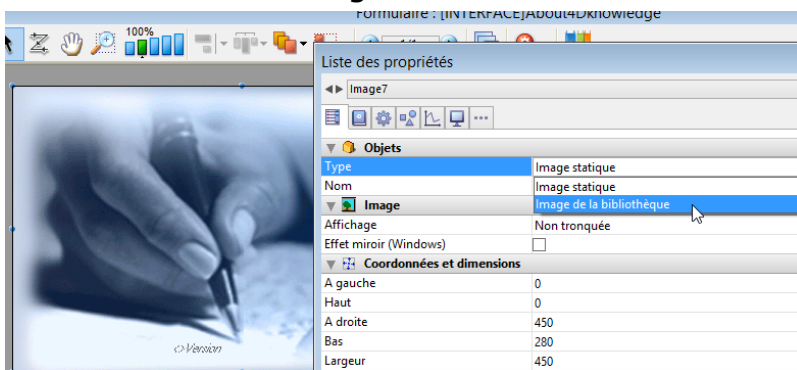
```
End if
```

```
End for
```

## Conversión de imágenes en estructura

Para hacer en versión 32 bits (*es decir*, antes de pasar a 64 bits).

1 - Transferir sus imágenes estáticas en los formularios a la librería de imágenes:



2 - Una vez sus imágenes sean transferidas, conviértalas a .png o .jpeg:

```
C_LONGINT($i;$SOA;$RIS;$PictRef)
```

```
C_TEXT($PictName)
```

```
C_PICTURE($Pict)
```

```
//----- inicialización de arrays -----
```

```
ARRAY LONGINT($aL_PictRef;0)
```

```
ARRAY TEXT($aT_PictName;0)
```

```
ARRAY TEXT($aT_Codecs;0)
```

```
PICTURE LIBRARY LIST($aL_PictRef;$aT_PictName)
```

```
$SOA:=Size of array($aL_PictRef)
```

```
//----- conversión de imágenes PICT a png -----
```

```
If($SOA>0)
```

```
    For($i;1;$SOA) // para cada imagen
```

```

    $PictRef:=$aL_PictRef{$i}
    $PictName:=$aT_PictName{$i}
    GET PICTURE FROM LIBRARY($aL_PictRef{$i};$Pict)
    GET PICTURE FORMATS($Pict;$at_Codecs)
    For($j;1;Size of array($at_Codecs))
        If($at_Codecs{$j}=".pict") // si el formato es obsoleto
            CONVERT PICTURE($Pict;".png") // conversión a png
// y almacenar en la librería
            SET PICTURE TO LIBRARY($Pict;$PictRef;$PictName)
        End if
    End for
End for
Else
    ALERT("The image library is empty.")
End if
//----- fin del método -----

```

## Aplicaciones fusionadas: utilización de los parámetros regionales

Ver **Lenguaje para los comandos y constantes**. En el contexto del despliegue de aplicaciones fusionadas, para usar los parámetros regionales puede editar el contenido del archivo de preferencias de 4D v1x en cada máquina local y definir la llave **"use\_localized\_language"** como **"true"**. Por ejemplo:

```

$UserPreference:=Get 4D folder(Active 4D Folder)+"4D Preferences v17.4DPreferences"
$ref:=DOM Parse XML source($UserPreference;True)
$refElem:=DOM Find XML
element($ref;"preferences/com.4d/method_editor/options";arrElementRefs) // Leer el
valor actual
DOM GET XML ATTRIBUTE BY NAME($refElem;"use_localized_language";$attrValue)
if($attrValue="false") // Vuelve al funcionamiento <v15
    DOM SET XML ATTRIBUTE($refElem;"use_localized_language";"true")
End if
DOM EXPOR TO FILE($ref;$UserPreference)
DOM CLOSE XML($ref)

```