









4D View Pro

-  Presentación
-  Funcionalidades de list box avanzadas
-  Creación de un área 4D View Pro
-  Utilización de un área 4D View Pro
-  Conversión de documentos 4D View
-  Fórmulas 4D View Pro
-  Lenguaje 4D View Pro
-  Lista alfabética de los comandos

Presentación

Sobre 4D View Pro

4D View Pro ofrece un conjunto de funcionalidades avanzadas relacionadas con las hojas de cálculo y las presentaciones capacidades en lista. 4D View Pro ofrece a los usuarios de 4D una alternativa moderna e integrada a algunas de las funcionalidades del producto de antigua generación 4D View.

4D View Pro se compone de dos partes:

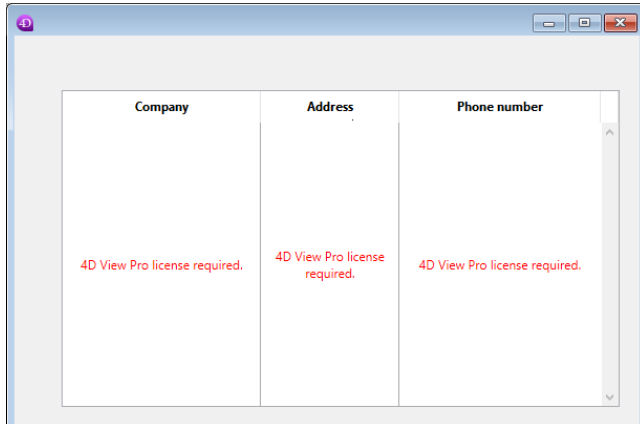
- Un conjunto de funcionalidades list box avanzadas dan un control completo a los desarrolladores sobre la altura de las líneas,
- Un componente y un área de formulario 4D que permite a los desarrolladores incluir una hoja de cálculo en sus formularios (*vista previa en 4D v16 R4*).

Instalación y activación

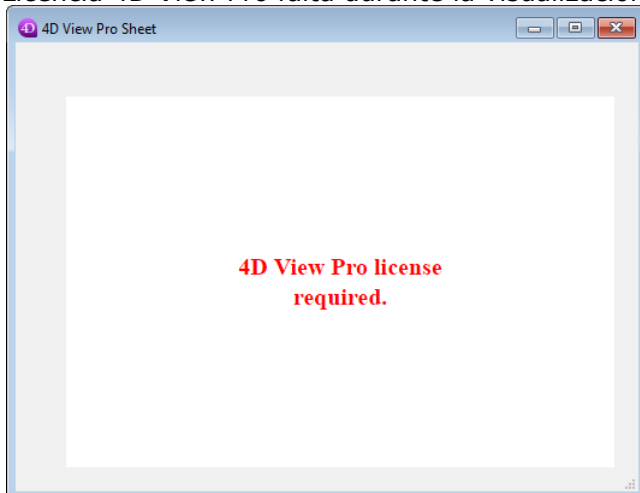
A diferencia del producto 4D View, las funcionalidades de 4D View Pro se incluyen directamente en 4D, lo que facilita su despliegue y el mantenimiento. No requiere instalación adicional.

Sin embargo, 4D View Pro requiere la misma licencia que 4D View. Necesita tener esta licencia instalada en su aplicación para poder utilizar estas funcionalidades. Cuando no se instala la licencia 4D View, el contenido de un objeto que requiere una funcionalidad 4D View Pro (list box o área 4D View Pro) no se muestra en la ejecución y en su lugar, aparece un mensaje de error.

- Licencia 4D View Pro falta mientras muestra un list box que utiliza una funcionalidad 4D View Pro:



- Licencia 4D View Pro falta durante la visualización de un área de hoja de cálculo de 4D View Pro:



Funcionalidades de list box avanzadas

Como se indica en la sección **Presentación**, una parte de 4D View Pro consiste en un conjunto de funcionalidades de list box avanzadas. Estas funcionalidades incluyen:

- Arrays objeto asociados con una columna de list box:

Atributo	Valor
Nombre del documento	MiReporte
Tipo del documento	PDF
Referencia	123456
Categoría	
Resumen incluido	<input checked="" type="checkbox"/>
Tamaño de área imprimible(Alto)	297 mm
Tamaño de área imprimible(Ancho)	210 mm
Vista preliminar	Visualizar

Para más información, consulte la página **Utilizar arrays objetos en las columnas (4D View Pro)**.




- Control de altura de línea variable en los list box::

Fila	Países	Población
1	Luxemburgo	502 202
2	Letonia	1 973 700
3	Kuwait	4 044 500
4	Croacia	4 284 889
5	Dinamarca	5 699 220
6	Nicaragua	6 071 045
7	Serbia	7 306 677
8	Honduras	8 249 574
9	Austria	8 572 895
10	Hungría	10 005 000
11	Republica Checa	10 674 947

Esta funcionalidad se basa en:

- Propiedad **Nueva propiedad**
 - Comandos **LISTBOX Get row height** y **LISTBOX SET ROW HEIGHT**
 - Constante **lk row height array** para los comandos **LISTBOX Get array** y **LISTBOX SET ARRAY**.
- Para más información consulte la documentación 4D.

- Altura de fila variable automática:

Picture	Name	Short text	Full text
	ABORT	The ABORT command is used from within an error-handling project method installed using the command ON ERR CALL .	If you use an <i>error-handling</i> project method to catch errors, 4D neither displays its standard error dialog box nor interrupts the execution of your code. Instead, 4D calls your error-handling project method (that you can see as an <i>exceptio handler</i>), and resumes the execution to the next line of code.
	ASSERT	The ASSERT command evaluates the <i>boolExpression</i> assertion passed in parameter and if it returns false, stops the code execution with an error message. The command works in interpreted and compiled mode.	If <i>boolExpression</i> is true, nothing happens. If it is false, the command triggers the error -10518 and displays by default the text of the assertion preceded by the message "Assert failed". You can intercept this error via a method installed using the ON ERR CALL command, in order, for example, to provide info for a log file. Optionally, you can pass a <i>messageText</i> parameter to display a custom error message instead of the text of the assertion. An assertion is an instruction inserted in the code that is responsible for detecting any anomalies. It therefore allows the use of an assertion during the evaluation of a condition (see the example). For more information about the operation of assertions and the parameters of this command please refer to the description of the ASSERT command.
	Asserted	The Asserted command has an operation similar to that of the ASSERT command, with one difference in that it returns	

Esta funcionalidad se basa en:

- Propiedad **Altura de fila automática** (disponible en los niveles de list box y columna)
- Constante **lk auto row height** para los comandos **LISTBOX SET PROPERTY** y **LISTBOX Get property**

- Comandos **LISTBOX SET AUTO ROW HEIGHT** y **LISTBOX Get auto row height** para definir límites de altura
Consulte la documentación 4D para mayor información.

Creación de un área 4D View Pro

Presentación

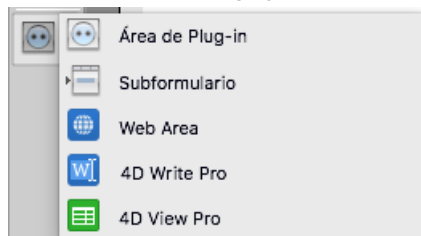
4D View Pro le permite insertar y mostrar un área de hoja de cálculo en sus formularios 4D. Una hoja de cálculo es una aplicación que contiene una cuadrícula de celdas en la que puede introducir información, ejecutar cálculos o mostrar imágenes.

Una vez que utilice las áreas de 4D View Pro en sus formularios, puede importar y exportar hojas de cálculo con los comandos 4D View Pro.

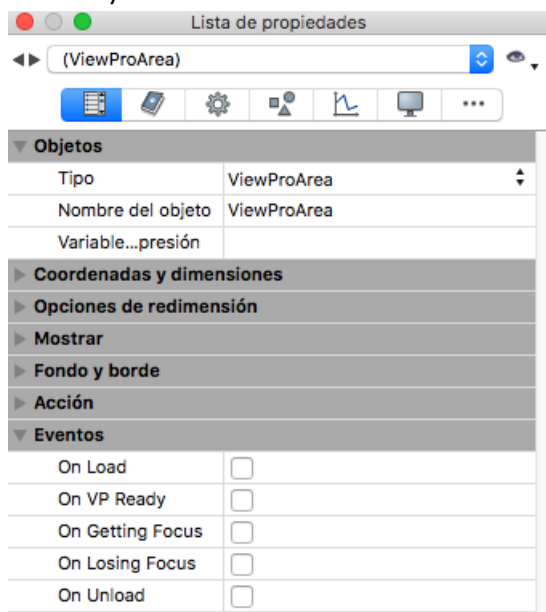
Compatibilidad: las áreas 4D View Pro no se pueden utilizar en versiones 32 bits de 4D.

Creación del área

Los documentos de 4D View Pro se muestran y editan manualmente en un objeto de formulario 4D denominado **4D View Pro**. Este objeto está disponible como parte de la última herramienta (Área de Plug-in, Área Web, etc.) que se encuentra en la barra de objetos:



Un área de formulario 4D View Pro se configura mediante propiedades estándar que se encuentran en la Lista de propiedades, como **Nombre de objeto** y **Variable o Expresión**, **Coordenadas**, **Pantalla**, **Acción** y **Eventos**.



Todos los temas

- **Nombre del objeto:** nombre del área de formulario 4D que contiene y muestra el documento 4D View Pro.
- **Variable o Expresión:** nombre de la variable de objeto de formulario del área 4D View Pro.

Cuando se ejecuta el formulario, el área 4D View Pro muestra una hoja de cálculo por defecto:



	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							

Utilización de un área 4D View Pro

Presentación

Cuando se ejecutan en formularios, las áreas de 4D View Pro ofrecen funcionalidades básicas de hoja de cálculo, incluyendo la edición de celdas y la introducción de fórmulas. Las funcionalidades más avanzadas están disponibles vía el lenguaje 4D View Pro.

Principios de base de selección, entrada y navegación

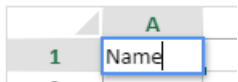
Las hojas de cálculo se componen de líneas y de columnas. Se asocia un número a cada fila. Una letra (o grupo de letras una vez que el número de filas sobrepasa el número de letras en el alfabeto) se asocia con cada columna. La intersección de una fila y una columna hace una celda. Las celdas pueden ser seleccionadas y su contenido editado.

Selección

- Para seleccionar una celda, simplemente haga clic en ella o use las flechas de dirección en el teclado. Su contenido (o fórmula) se muestra dentro de la celda.
- Para seleccionar varias celdas continuas, arrastre el ratón de un extremo a otro de la selección. También puede hacer clic en los dos extremos de la selección mientras mantiene pulsada la tecla **Mayús**.
- Para seleccionar todas las celdas de la hoja de cálculo, haga clic en la celda en la parte superior izquierda del área.
- Para seleccionar una columna, haga clic en la letra correspondiente (o conjunto de letras).
- Para seleccionar una fila, haga clic en el número correspondiente.
- Para seleccionar un grupo de celdas que no sean continuas, mantenga presionada la tecla **Ctrl** (Windows) o la tecla **Comando** (Mac) y haga clic en cada celda que desea seleccionar.
- Para anular la selección de celdas, simplemente haga clic en cualquier lugar dentro de la hoja de cálculo.

Entrada y navegación

Hacer doble clic en una celda permite pasar al modo de entrada en la celda correspondiente. Si la celda no está vacía, el cursor de inserción se coloca después del contenido de la celda.



Los datos se pueden introducir directamente una vez que una celda ya esté seleccionada, incluso si el cursor de inserción no está visible. La entrada reemplaza entonces el contenido de la celda.

La tecla **Tab** valida la entrada de celda y selecciona la celda a su derecha. La combinación de las teclas **Mayús + Tab** valida la entrada de la celda y selecciona la celda a su izquierda.

La tecla **Retorno de carro** valida la entrada de la celda y selecciona la celda debajo de ella. La combinación de las teclas **Mayús + Retorno de carro** valida la entrada de la celda y selecciona la celda por encima.

Las teclas de dirección (flechas) permiten mover una celda en la dirección indicada por la flecha.

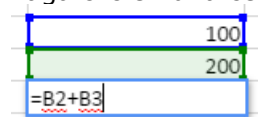
Entrada de una fórmula, de una función o de una referencia

Para introducir una fórmula o una función en un área 4D View Pro:

1. Seleccione la celda en la que ingresará la fórmula o función.
2. Introduzca = (el signo igual).
3. Introduzca la fórmula.

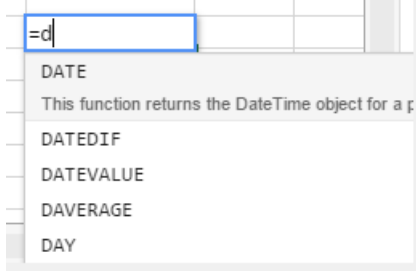
O

Haga clic en una celda para introducir su referencia en la fórmula.



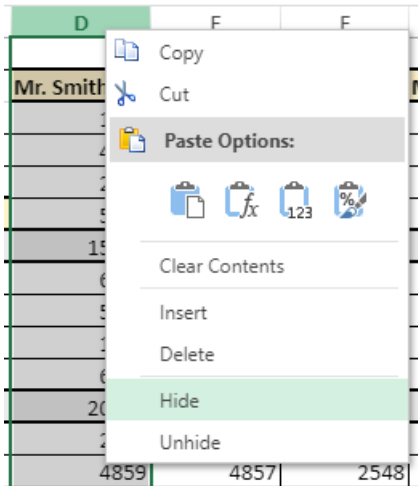
O

Escriba la primera letra de la función a introducir. Aparece un menú emergente con las funciones y referencias disponibles, que le permite seleccionar los elementos deseados:



Menú Contextual

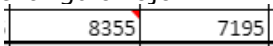
Las áreas 4D View Pro se benefician de un menú contextual automático que ofrece a los usuarios funciones de edición estándar como copiar y pegar, pero también con funciones básicas de hoja de cálculo:



Nota: las funcionalidades Copiar/Cortar y Pegar del menú contextual solo funcionan dentro del área de hoja de cálculo, no tienen acceso al panel de sistema. Sin embargo, los accesos directos del sistema, como **Ctrl+c/Ctrl+v** funcionan y se pueden usar para intercambiar datos entre el área y otras aplicaciones.

Este menú ofrece funcionalidades adicionales según el área en la que se haga clic:

- haga clic en una columna o encabezado de línea: **Insertar**, **Eliminar**, **Ocultar** o **Mostrar** los contenidos
- haga clic en una celda o en un rango de celdas:
 - **Filtro:** permite ocultar líneas por medio de filtros (ver [Filtrar filas](#) en la documentación de SpreadJS).
 - **Ordenar:** ordena el contenido de la columna.
 - **Insertar comentario:** permite al usuario ingresar un comentario para un área. Cuando se ingresa un comentario para un área, la celda superior izquierda del área muestra un pequeño triángulo rojo:



Conversion de documentos 4D View

Puede convertir sus documentos 4D View en áreas 4D View Pro utilizando el comando **VP Convert from 4D View**. La mayoría de las propiedades e información almacenadas en documentos 4D View se convierten automáticamente, incluidos formatos, estilos, bordes, valores, fórmulas, selecciones, zoom, etc. En general, los documentos de 4D View convertidos se renderizarán en las áreas 4D View Pro exactamente como se representaron en áreas de 4D View, como las siguientes:

Documento original de 4D View:

Client_Desc_origine

NOTE D'HONORAIRES
Suivant contrat de 1405/12
payable le 13/06/12

ÉcoLogis
88 rue du Bout du Monde
88058 - Echoubaoulin

PRO_12_024 - Villa de M. et Mme BEAUCHEMIN
Construction d'une maison individuelle

BARZAC
85 rue Boulangers
45100 LA SOURCE

Éléments de missions	%	Montant HT
Avant Projet Détaillé	20.00 %	4 180.00 €
Dossier de Permis de Construire	23.00 %	4 807.00 €
Assistance à la passation des contrats de travail	15.00 %	3 135.00 €
Direction de l'exécution des contrats de Travaux	34.00 %	7 106.00 €
Assistance aux Opérations de Réception	8.00 %	1 672.00 €
TOTAL	100.00 %	20 900.00 €

Avancement	
Taux	Montant HT
100.00 %	4 180.00 €
100.00 %	4 807.00 €
100.00 %	3 135.00 €
46.15 %	3 279.72 €
	15 401.72 €

Total des missions réalisées HT : 15 401.72 €
Moins précédemment facturé HT : 14 855.10 €
Montant de la révision HT : 546.62 €

Montant de la note HT : **546.62 €**
TVA à 19.6% : **107.14 €**
Montant TTC : **653.76 €**

Versements directs aux sous-traitants : 104.60 €
Montant TTC à payer : **549.16 €**

Merci d'indiquer le numéro de facture ci-dessus sur votre ordre de paiement ou au dos de votre chèque.
TVA acquittée sur les encaissements - Exonérée 0% - Pénalité pour retard : 3 fois le taux d'intérêt légal en vigueur
SIRET 400 958 879 00065 - Code APE 742A - N° d'identification intracommunautaire - FR 49 600 932 884

Documento convertido en área 4D View Pro

4D View Pro area

NOTE D'HONORAIRES
Suivant contrat de 1405/12
payable le 13/06/12

ÉcoLogis
88 rue du Bout du Monde
88058 - Echoubaoulin

PRO_12_024 - Villa de M. et Mme BEAUCHEMIN
Construction d'une maison individuelle

BARZAC
85 rue Boulangers
45100 LA SOURCE

Éléments de missions	%	Montant HT
Avant Projet Détaillé	20.00 %	4 180.00 €
Dossier de Permis de Construire	23.00 %	4 807.00 €
Assistance à la passation des contrats de travail	15.00 %	3 135.00 €
Direction de l'exécution des contrats de Travaux	34.00 %	7 106.00 €
Assistance aux Opérations de Réception	8.00 %	1 672.00 €
TOTAL	100.00 %	20 900.00 €

Avancement	
Taux	Montant HT
100.00 %	4 180.00 €
100.00 %	4 807.00 €
100.00 %	3 135.00 €
46.15 %	3 279.72 €
	15 401.72 €

Total des missions réalisées HT : 15 401.72 €
Moins précédemment facturé HT : 14 855.10 €
Montant de la révision HT : 546.62 €

Montant de la note HT : **546.62 €**
TVA à 19.6% : **107.14 €**
Montant TTC : **653.76 €**

Versements directs aux sous-traitants : 104.60 €
Montant TTC à payer : **549.16 €**

Merci d'indiquer le numéro de facture ci-dessus sur votre ordre de paiement ou au dos de votre chèque.
TVA acquittée sur les encaissements - Exonérée 0% - Pénalité pour retard : 3 fois le taux d'intérêt légal en vigueur
SIRET 400 958 879 00065 - Code APE 742A - N° d'identification intracommunautaire - FR 49 600 932 884

Hacemos todo lo posible para garantizar que los documentos convertidos permanezcan lo más fiel posible al original, pero es posible que algunas funciones no se rendericen por completo. Estas se describen en el párrafo a continuación .

Proceso de conversión

Nota: la conversión de documentos 4D se mejora continuamente, con base en los comentarios de los usuarios. Se recomienda encarecidamente guardar siempre una copia de los BLOBs 4D View o documentos

originales, incluso después de una conversión exitosa.

Con respecto al estado actual de su documento de plug-in 4D View, el proceso de conversión requiere los siguientes pasos.

1. Cargue su documento 4D View (.4pv) en un BLOB:

Nota: si su documento 4D View ya está almacenado en un campo BLOB, vaya al paso 2.

```
C_BLOB($pvblob)
DOCUMENT TO BLOB("document.4PV";$pvblob)
```

2. Llame a **VP Convert from 4D View** con el BLOB que contiene el documento 4D View:

```
C_OBJECT($vpObj)
$vpObj:=VP Convert from 4D View($pvblob)
```

3. Asigne el objeto resultante a un objeto de formulario de área de 4D View Pro o a un documento para ver los resultados.

```
VP IMPORT FROM OBJECT("4DViewProArea";$vpObj)
```

Detalles de la conversión

La siguiente tabla ofrece el estado actual de las áreas de conversión principales. Tenga en cuenta que esta lista se actualizará regularmente, ya que el proceso de conversión se mejora continuamente.

Funcionalidad	Estado de conversión	Comentarios
Atributos del documento	<p>Los atributos de visualización del documento se convierten: celdas seleccionadas, zoom, visualización de cuadrícula.</p> <p>La información del documento se convierte: versión, título, autor del tema, empresa, nota, creación y fecha de modificación.</p>	
Columnas y filas	<p>Todas las columnas y filas definidas se convierten con su tamaño original.</p> <p>Los encabezados de columnas y filas se convierten sin restricciones.</p>	
Bordes	<p>Los bordes se convierten con su grosor y color. Los ajustes de la pantalla de cuadrícula se convierten.</p>	<p>Solo hay disponibles un borde de barra y un borde de barra doble en 4D View Pro. Los bordes de una sola barra se convierten de la misma forma que el original, cualquier borde de doble barra se convierte al modelo de doble barra 4D View Pro.</p>
Separadores	<p>Actualmente no convertido</p>	
Estilos y fuentes	<p>Los estilos y las hojas de estilo se convierten. Los estilos condicionales no son compatibles.</p> <p>Los estilos de rotación (texto orientado) actualmente no están convertidos</p>	<p>Los estilos de texto obsoletos (es decir, Shadow, Condensed, etc.) y las fuentes QuickDraw no se convierten.</p>
Nombres de formatos y celdas	<p>Los formatos de celda se convierten a formatos disponibles con una representación similar. Todos los formatos de tipo de datos son compatibles: texto, número, fecha y hora, booleano, imagen. Los nombres de celda se convierten.</p> <p>Los formatos 4D definidos por el usuario (comenzando por " ") actualmente no están convertidos</p> <p>Los controles creados con PV SET CELL CONTROL (botón / botón de opción / casilla de verificación / menú desplegable / cuadro combinado) actualmente no son compatibles</p>	<p>La conversión de formato podría dar resultados inesperados para algunos documentos porque los formatos a veces dependen de la configuración del sistema (es decir, para el separador decimal) y pueden haberse definido globalmente o por separado.</p>
Imágenes	<p>Las imágenes son compatibles y se convierten con algunas limitaciones</p> <p>Las imágenes con formato truncado centrado/replicado no están convertidas actualmente</p>	<p>El formato de imagen 4D View incluye varios códecs y está en desuso. Las imágenes convertidas conservan solo el códec más apropiado para la representación html (svg, png, jpeg, gif) y se guardan en base64.</p> <p>Las imágenes de fondo no se replican en cada página (este concepto no existe en 4D View Pro).</p>
Impresión	<p>Actualmente no compatible</p>	<p>Las opciones de impresión y las configuraciones de impresión definidas en el documento 4D View se convierten.</p>
Enlaces dinámicos	<p>Las celdas o columnas vinculadas a campos o variables no son compatibles actualmente</p>	
Fórmulas	<p>Las fórmulas se convierten pero las referencias a los comandos, métodos, variables y campos 4D no son compatibles actualmente</p>	

Fórmulas 4D View Pro

Generalidades


Las funciones 4D View Pro se utilizan en fórmulas. Cada fórmula de 4D View Pro es una expresión que devuelve un valor. Todas las expresiones se componen de operandos y operadores:

- Los **operadores** disponibles en 4D View Pro son generalmente similares a los de 4D (ver **Constantes y Operadores**)
- Los **operandos** se dividen en varias categorías:
 - constantes,
 - referencias a otras celdas (relativas, absolutas, mezclados o por nombre),
 - variables 4D, campos y funciones,
 - funciones 4D View Pro.

Para introducir una fórmula:

1. Seleccione la celda en la que ingresará la fórmula o función.
2. Intro= (el signo igual).
3. Escriba la fórmula y luego presione la tecla Intro

Esta sección lista las fórmulas esenciales de 4D View Pro, que también se ofrecieron en el plug-in 4D View heredado. Se pueden encontrar fórmulas adicionales compatibles con 4D View Pro en la [Documentación de hojas de cálculo](#).

-  Constantes y Operadores
-  Referencias de celdas
-  Conversión de fórmulas del plug-in 4D View
-  PI
-  ABS
-  ACOS
-  AND
-  ASIN
-  ATAN
-  AVERAGE
-  COLUMNLETTER
-  COS
-  COUNTA
-  EXP
-  FALSE
-  FINDCELL
-  FV
-  IF
-  INDIRECT
-  ISBLANK
-  LEN
-  LN
-  LOOKUP
-  MAX
-  MID
-  MIN
-  NOT
-  NOW
-  NPER
-  OR
-  PMT
-  PV
-  RAND
-  RATE

- ⚙ ROUND
- ⚙ ROW
- ⚙ SIN
- ⚙ SQRT
- ⚙ STDEV.P
- ⚙ SUBSTITUTE
- ⚙ SUM
- ⚙ TAN
- ⚙ TEXT
- ⚙ TODAY
- ⚙ TRUE
- ⚙ TYPE
- ⚙ VAR.P

✚ Constantes y Operadores

Operadores por tipo de datos

4D View Pro soporta cinco tipos de datos. Para cada tipo de datos, constantes y operadores específicos son soportados.

Tipos de datos	Constantes	Operadores
Número	1.2 1.2 E3 1.2E-3 10.3x	+ (adición) - (resta) * (multiplicación) / (división) ^ (exponente, el número de veces a multiplicar un número por sí mismo) % (porcentaje: divide el número antes del operador por cien)
Fecha	10/24/2017	+ (fecha + número de días -> fecha) + (fecha + hora -> fecha + hora del día) - (fecha - número de días -> fecha) - (fecha - fecha -> número de días entre los dos)
Hora	10:12:10	Operadores de duración: + (adición) - (resta) * (duración * número -> duración) / (duración / número -> duración)
Cadena	'Sophie' o "Sophie"	& (concatenación)
Booleano	TRUE o FALSE	-

Operadores de comparación

Los siguientes operadores se pueden usar con dos operandos del mismo tipo:

Operador	Comparación
=	igual a
<>	diferente de
>	mayor que
<	menor que
>=	mayor o igual a
<=	menor o igual que

Prioridad de los operadores

Lista de operadores de mayor a menor importancia:

1. ()
2. -
3. +
4. %
5. ^
6. * /
7. + -
8. &
9. = > < >= <= <>

Prioridad de los operandos en las fórmulas

Cuando dos o más operandos diferentes tienen el mismo nombre, 4D View Pro determina el tipo de cada elemento de acuerdo con el siguiente orden:

Prioridad	Tipo de elemento
1	Referencia de celda
2	Nombre de la celda
3	Función 4D View Pro
4	Método proyecto
5	Comando 4D
6	Variable

Referencias de celdas

Las fórmulas a menudo se refieren a otras celdas por direcciones de celda. Puede copiar estas fórmulas en otras celdas. Cuando copia o mueve estas fórmulas a ubicaciones nuevas, cada dirección de celda en esa fórmula cambiará o permanecerá igual, dependiendo de cómo se digite. Una referencia que cambia se denomina referencia relativa y se refiere a una celda según qué tan a la izquierda/derecha y arriba/abajo esté de la celda con la fórmula. Una referencia que siempre apunta a una celda particular se llama referencia absoluta.

También puede crear una referencia mixta que siempre apunta a una fila o columna fija. Por ejemplo, la siguiente fórmula, ingresada en la celda C8, agrega los valores en las dos celdas sobre ella y muestra el resultado.

= C6 + C7

Esta fórmula se refiere a las celdas C6 y C7. Es decir, se instruye a 4D View Pro para que haga referencia a estas otras celdas para los valores a usar en la fórmula.

Notación de referencias

Si usa solo coordenadas de celda, por ejemplo, C5, 4D View Pro interpreta la referencia como relativa. Puede hacer que la referencia sea una referencia absoluta colocando un signo de dólar delante de la letra y el número, como en \$C\$5.

Puede mezclar referencias absolutas y relativas insertando un signo de dólar delante de la letra o el número solo, por ejemplo, \$C5 o C\$5. Una referencia mixta le permite especificar la fila o la columna como absoluta, mientras permite que la otra parte de la dirección se refiera de manera relativa.

Una forma conveniente, rápida y precisa de especificar una referencia absoluta es nombrar la celda y usar ese nombre en lugar de la dirección de la celda. Una referencia a una celda con nombre es siempre absoluta.

La siguiente tabla muestra el efecto de las diferentes notaciones:

Ejemplo	Tipo de referencia	Descripción
C5	Relativa	La referencia es a la ubicación relativa de la celda C5, dependiendo de la ubicación de la celda en la que se utiliza por primera vez la referencia.
\$C\$5	Absoluta	La referencia es absoluta. Siempre se referirá a la celda C5 sin importar dónde se use.
\$C5	Mixta	La referencia es siempre a la columna C, pero la referencia de fila es relativa a la ubicación de la celda en la que se utiliza por primera vez la referencia.
C\$5	Mixta	La referencia es siempre a la fila 5, pero la referencia de la columna es relativa a la ubicación de la celda en la que se utiliza por primera vez la referencia.
Cell name	Absoluta	La referencia es absoluta. Siempre se referirá a la celda nombrada sin importar dónde se use la referencia.

✚ Conversión de fórmulas del plug-in 4D View

Generalidades

Como se indica en la página [Conversión de documentos 4D View](#), la mayoría de los contenidos y propiedades del documento del plug-in 4D View se pueden convertir en documentos 4D View Pro.

Las fórmulas también se convierten. Sin embargo, los lenguajes de fórmula de 4D View y 4D View Pro son algo diferentes. Tres casos de compatibilidad pueden ocurrir:

- una funcionalidad 4D View (operador, constante, función) es exactamente la misma en 4D View Pro: en este caso, la conversión es transparente.
- una funcionalidad 4D View es compatible con 4D View Pro a través de una función u operador diferente: en este caso, se realiza una asignación automática
- una funcionalidad 4D View es parcial o no es compatible con 4D View Pro: en este caso, será necesario adaptar sus fórmulas convertidas para que funcionen como se esperaba.

Las siguientes tablas listan las funcionalidades de la fórmula 4D View y sus funcionalidades correspondientes en 4D View Pro.

Operadores

	4D View	4D View Pro
Operadores numéricos		
Addition	+	+
Substraction	-	-
Multiplication	*	*
Division	/	/
Remainder	\	MOD
Integer division	÷	TRUNC(a/b)
Exponent	^	^
Percentage	%	%
Operadores Booleanos		
AND	&	AND
OR		OR
Not	~	NOT
Operadores cadena		
concatenation	+	&
destruction	-	SUBSTITUTE , ej.: "Down Trend"- "Down" se remplaza por SUBSTITUTE("Down Trend", "Down", "")
position	\	FIND (sensibles a mayúsculas o minúsculas) o SEARCH (case insensitive)
Operadores Fecha		
date+days->date	+	+
date+time->date+time of day	+	+
date-days->date	-	-
date-date->number of days	-	-
Operadores duración		
addition	+	+
substraction	-	-
multiplication	*	*
division	/	/
Operadores comparación		
equality	=	=
difference	#	<>
greater than	>	>
less than	<	<
greater than or equal to	>=	>=
less than or equal to	<=	<=

Funciones

4D View	4D View Pro	Comentario
Abs	ABS	
AddToDate	DATE	AddToDate(date;years;months;days) se reemplaza por DATE (<u>YEAR</u> (date)+years, <u>MONTH</u> (date)+months, <u>DAY</u> (date)+days)
date+time	TIME	DATE (date) + TIME (time)
And	AND	
ArcCos	ACOS	
ArcSin	ASIN	
ArcTan	ATAN	
Area	-	n/a (no plug-in area)
Average	AVERAGE	
Cell	INDIRECT	
Column	COLUMNLETTER	<u>COLUMN</u> devuelve un número (no una letra)
Cos	COS	
Count	COUNTA	
CurrentDate	TODAY	
CurrentTime	NOW	
CVCompound	PV	CVCompound(1%;5;1000) se reemplaza por PV (1%,5,-1000)
CVSimple	PV	CVSimple(1%;5;5*1000) se reemplaza por PV (1%,5,, -1000) -- note the two consecutive commas
Empty	ISBLANK	
Eval4D	-	Currently not available
Exp	EXP	
False	FALSE	
Find	LOOKUP	
FindCell	FINDCELL	
FVCompound	FV	FVCompound(1%;35;35*1000) se reemplaza por FV (1%,35,-1000)
FVSimple	FV	FVSimple(12%;35;35*1000) se reemplaza por FV (12%,35,, -35*1000) -- note las dos comas consecutivas
If	IF	
Length	LEN	
Log	LN	
Max	MAX	
Min	MIN	
MonthlyValue	PMT	MonthlyValue(10.5%/12,48,6500) se reemplaza por PMT (10.5%/12,48,-6500)
Not	NOT	
Or	OR	
PeriodNumber1	NPER	PeriodNumber1(10.5%/12;166.42;6500) se reemplaza por NPER (10.5%/12,-166.42,6500)
PeriodNumber2	NPER	PeriodNumber2(10.5%/12,5000,3000) se reemplaza por NPER (10.5%/12,,3000,-5000) -- note las dos comas consecutivas
Pi	PI	
Random	RAND	
Range	INDIRECT	Range("A1";"A3") se reemplaza por INDIRECT ("A1:A3") -- note el punto y coma entre A1 y A3
Rate1	RATE	Rate1(5;1000;3000) se reemplaza por RATE (5,-1000,3000)
Rate2	RATE	Rate2(5,6000,2800) se reemplaza por RATE (5,,2800,-6000) -- note las dos comas consecutivas
Rounding	ROUND	
Row	ROW	
Sin	SIN	
SquareRoot	SQRT	
StdDeviation	STDEV.P	
String	TEXT	
SubString	MID	
Sum	SUM	
Tan	TAN	

True
Type
Variance

TRUE
TYPE
VAR.P

Tipos devueltos en 4D View Pro son diferentes de 4D View

PI

PI ()

Este comando no requiere parámetros

Descripción

La **PI** función devuelve el valor de Pi como 3.14159265358979.

ABS

ABS (valor)

Parámetro	Tipo	Descripción
valor	Real, Expresión	→ Número cuyo valor absoluto es devuelto

Descripción

La función **ABS** calcula el (los) valor(es) absoluto(s) del *valor* especificado. Si el *valor* es negativo, se devolverá un valor positivo. Acepta datos numéricos como valores o expresiones y devuelve datos numéricos.

Ejemplo

```
ABS(-6) //result:= 6
```

```
ABS(16-26) //result:= 10
```

```
ABS(6) //result:= 6
```

ACOS

ACOS (valor)

Parámetro	Tipo	Descripción
valor	Real	→ Ángulo cuyo arcocoseno se devuelve. Debe estar entre -1 y +1.

Descripción

La función **ACOS** calcula el ángulo del arcocoseno especificado en *valor*. *valor* debe incluirse en el rango de -1 a +1.

El ángulo devuelto está en radianes entre 0 y PI. Para convertir el resultado en grados, multiplique el resultado por 180/PI.

Ejemplo

```
ACOS(0.5) //result:= 1.0471975512
```

AND

AND (valorLogico {, valorLogico2 , ... , valorLogicoN})

Parámetro	Tipo	Descripción
valorLogico	Booleano, Número, Expresión	→ Valor(es) a evaluar

Descripción

La función **AND** devuelve TRUE si todos los argumentos son verdaderos; de lo contrario, devuelve FALSE si al menos un argumento es falso.
Acepta valores booleanos como numéricos (0 o 1) o expresiones lógicas (TRUE o FALSE) hasta 255 argumentos. También puede especificar un único array en lugar de listar los valores por separado, o hasta 255 arrays. También puede especificar el *valorLogico* como una expresión.

Ejemplo

```
AND(D12,E12)
```

```
AND(D2:D12)
```

```
AND(5+3=8,5+1=6) //TRUE
```

```
AND(1,TRUE) //TRUE
```


ASIN

ASIN (valor)

Parámetro	Tipo		Descripción
valor	Real	→	Seno del ángulo. Debe estar entre -1 y +1.

Descripción

La función **ASIN** calcula el arcoseno, el ángulo cuyo seno se especifica en *valor*. Acepta y devuelve datos numéricos.

En *valor*, especifique el seno del ángulo. El seno debe ser un valor entre -1 y +1.

El ángulo se devuelve en radianes entre $-\pi/2$ y $\pi/2$. Para convertir el resultado en grados, multiplíquelo por $180/\pi$.

Ejemplo

```
ASIN(0.5) //0.5235987756
```

ATAN

ATAN (valor)

Parámetro	Tipo	Descripción
valor	Real →	Tangente de un ángulo. Debe estar entre -1 y +1.

Descripción

La función ATAN calcula el arcotangente, es decir, el ángulo cuya tangente se especifica en *valor*. Acepta y devuelve datos numéricos.

En *valor*, especifique la tangente del ángulo a devolver. Debe estar entre -1 y +1.

El ángulo se devuelve en radianes entre $-\pi/2$ y $\pi/2$. Para convertir el resultado en grados, multiplique el resultado por $180/\pi$.

Ejemplo

```
ATAN(1) //resultado:= 0.7853981634
```

AVERAGE

AVERAGE (valor {, valor2 , ... , valorN})

Parámetro	Tipo	Descripción
valor	Real, Array	→ Número(s) cuya media debe ser calculada

Descripción

La función **AVERAGE** calcula el promedio del *valor*.

En *valor*, puede pasar:

- valores reales o enteros largos,
- un rango o varios rangos de celdas (array).

Se pueden incluir hasta 255 argumentos.

Ejemplo

```
AVERAGE(98,72,85) //result=85
```

```
AVERAGE(A1,B3,D5,E9,L8,L9)
```

```
AVERAGE(R1C1,R3C2)
```

```
AVERAGE(A1:A9)
```

```
AVERAGE(A1:A9,B1:B9,D5:D8)
```

COLUMNLETTER

COLUMNLETTER ({referencia})

Parámetro	Tipo		Descripción
referencia	CellRef	→	Una celda o rango de celdas

Descripción

La función **COLUMNLETTER** devuelve la letra de la columna de *referencia*. *referencia* puede ser una celda o un rango de celdas. Si se omite el argumento de *referencia*, el argumento predeterminado es la referencia de la celda en la que se coloca la función **COLUMNLETTER**.

Ejemplo

```
COLUMNLETTER(A9) //A
```

```
COLUMNLETTER(B1:B5) //B
```

COS

COS (valor)

Parámetro	Tipo	Descripción
valor	Real	Ángulo cuyo coseno es devuelto

Descripción

La función **COS** devuelve el coseno del ángulo especificado en *valor*. Acepta y devuelve datos numéricos. En *valor*, pase todo número real (un ángulo) por el cual devolver el coseno. Debe expresarse en radianes. Si el ángulo está en grados, multiplíquelo por $\text{PI}/180$ para convertirlo a radianes.

Ejemplo

```
COS(45*PI()/180) //0.7071067812
```

COUNTA

COUNTA (valor {, valor2 , ... , valorN})

Parámetro	Tipo	Descripción
valor	Real, Array	→ Celdas o rango de celdas a contar

Descripción

La función **COUNTA** cuenta la cantidad de celdas especificadas en *valor* que no están vacías (es decir, celdas que contienen números, texto o valores lógicos). Acepta referencias de celda y devuelve datos numéricos.

En *valor*, puede pasar hasta 255 celdas separadas o un único array de valores.

Ejemplo

```
COUNTA(B2,D2,E4,E5,E6)
```

```
COUNTA(A1:G5)
```

EXP

EXP (valor)

Parámetro	Tipo		Descripción
valor	Número	→	Número a evaluar



Descripción

La función **EXP** devuelve la base log natural ($e = 2.71828\dots$) elevada a la potencia del número especificado en *valor*. Acepta y devuelve datos numéricos.

Esta función es la inversa de **LN**, por lo que $\text{EXP}(\text{LN}(x))$ da como resultado x .

Ejemplo

```
EXP(B3)
```

```
EXP(1) //2.17828...
```

FALSE

FALSE ()

Este comando no requiere parámetros

Descripción

La función **FALSE** devuelve el valor lógico FALSE (0).

Ejemplo

```
NOT(FALSE) //TRUE
```


FINDCELL

FINDCELL (*aBuscar* , *rangoaBuscar*)

Parámetro	Tipo		Descripción
<i>aBuscar</i>	CellRef	→	Valor a buscar
<i>rangoaBuscar</i>	CellRef	→	Celdas a buscar

Descripción

La función **FINDCELL** busca el valor *aBuscar* en el rango *rangoaBuscar* de celdas y devuelve la referencia de la celda en la que se encontró. Esta referencia no se puede mostrar, pero puede ser utilizada por otras funciones de 4D View Pro que aceptan una referencia de celda (*CellRef*) como parámetro.

aBuscar debe contener la referencia de una celda que en realidad contiene el valor a encontrar.

Ejemplo

Asumiendo que celda C3 contiene 10:

```
FINDCELL(C3,A1:B9) //devuelve 10 si el valor se encuentra realmente en el rango de celda A1: B9; de lo contrario, devuelve un error.
```

$FV(i, n, m \{, f\})$

Parámetro	Tipo	Descripción
i	Número	→ La tasa de interés por un período
n	Número	→ Número de periodos
m	Número	→ Para interés compuesto: el pago mensual al final de cada período (use un valor negativo). Para interés simple: pase un parámetro vacío (ver el ejemplo)
f	Número	→ Para tasa de interés única: el valor final al final de un período (use un valor negativo)

Descripción

La función **FV** calcula:

- el valor final de una suma utilizando interés compuesto, o
- el valor final de una suma utilizando interés simple

Para calcular el valor adquirido durante una inversión, si los pagos mensuales se pagan al final del período, pase el parámetro m y omita el parámetro f . Aquí está la fórmula para este cálculo:

$$FV(i,n,m) = m \times \frac{(1+i)^n - 1}{i}$$

Para calcular el valor final de una suma con interés simple, pase el parámetro f y pase un parámetro vacío (,,) para el marcador de posición m . Aquí está la fórmula para este cálculo:

$$FV(i,n,,f) = f \times (1+i)^n$$

Ejemplo 1

Interés compuesto: usted planea depositar €1,000 cada mes en una cuenta de ahorros, que le genera un interés anual del 12%, durante 35 meses.

`FV(12%,35,-1000) //41660.275603126`

Ejemplo 2

Tasa de interés simple: el mismo tipo de escenario que el anterior.

`FV(12%,35,,-35*1000) //1847986.69`

IF

IF (valorPrueba , valorTrue , valorFalse)

Parámetro	Tipo		Descripción
valorPrueba	Expresión	→	Valor o expresión a evaluar
valorTrue	Expresión	→	Valor a devolver si la prueba se evalúa como verdadera
valorFalse	Expresión	→	Valor a devolver si la prueba se evalúa como falsa o 0

Descripción

La función **IF** realiza una comparación y devuelve uno de los dos valores ofrecidos según esa comparación. Acepta datos numéricos (booleanos) y devuelve todo tipo de datos.

El valor de *valorPrueba* se evalúa. Debe ser o evaluar datos numéricos, donde valores distintos de cero indican TRUE, y un valor de cero indica FALSE. Puede contener uno de los operadores relacionales: mayor que (>), menor que (<), igual a (=) o no igual a (<>). Si *valorPrueba* es:

- no es cero (TRUE), entonces se devuelve *valorPrueba*.
- cero (FALSE), luego se devuelve *valorFalse*.

Ejemplo

Desea evaluar B1, dando el valor de las ventas.

```
IF(B1 < 200, "Declining result", "Good result") // "Good result" se escribe si B1 > 200
```

INDIRECT

INDIRECT (celda | rangoCelda)

Parámetro	Tipo	Descripción
celda rangoCelda	CellRef →	Referencia a una celda, un nombre definido como una referencia, o una referencia de cadena de texto a una celda o a un rango

Descripción

La función **INDIRECT** devuelve el contenido de *celda*. El parámetro *celda* (obligatorio) puede ser cualquier referencia de celda, incluida la referencia absoluta, como \$A\$1 o una cadena de caracteres. Use **INDIRECT** cuando desee cambiar la referencia a una celda dentro de una fórmula sin cambiar la fórmula en sí.

La función **INDIRECT** también puede devolver la referencia interna de un rango de celdas (la referencia no se puede mostrar pero puede ser utilizada por otras funciones de 4D View Pro).

Ejemplo

```
INDIRECT("A1") //devuelve el contenido de celda A1
```

```
COLUMN(INDIRECT("A1:A3")) //columna 1  
ROW(INDIRECT("A1:F1")) //fila 1
```

ISBLANK

ISBLANK (valor)

Parámetro	Tipo	Descripción
valor	CellRef, Expresión, Número, Texto	→ Valor a evaluar

Descripción

La función **ISBLANK** prueba si el contenido de una celda, un número, un texto o cualquier expresión está vacío. Esta función devuelve TRUE si el valor se refiere a una celda vacía o a ningún dato.

Nota: las celdas que contienen una cadena vacía ("") se consideran en blanco.

Ejemplo

```
IF(ISBLANK(A1);"Error";0) //"Error" si celda A1 está vacía
```

```
ISBLANK(B1)
```

```
ISBLANK(A4-52)
```

```
ISBLANK(4) //FALSE
```

LEN

LEN (valor)

Parámetro	Tipo	Descripción
valor	Texto, CellRef	→ Texto cuya longitud de caracteres contar

Descripción

La función **LEN** devuelve el número de caracteres en la cadena *valor*.

En *valor*, pase el texto cuya longitud desea encontrar. Debe ser una cadena o una referencia de celda a un valor de cadena.

Nota: los espacios cuentan como caracteres

Ejemplo

```
LEN("4D, Inc.") //8
```

LN

LN (valor)

Parámetro	Tipo	→	Descripción
-----------	------	---	-------------

valor

Número



Número mayor de 0 a evaluar

Descripción

La función **LN** devuelve el logaritmo natural de *valor*. Acepta y devuelve datos numéricos. En *valor*, pase un número positivo (mayor que cero).

Nota: **LN** es el inverso de **EXP**, por lo que LN(EXP(x)) es x.

Ejemplo

```
LN(10) //2.30258509...
```

```
LN(EXP(1)) //1
```

LOOKUP

LOOKUP (aBuscar , intervaloABuscar , intervaloRetorno)

Parámetro	Tipo	Descripción
aBuscar	CellRef, Texto, Número, Booleano	➔ Valor a buscar
intervaloABuscar	CellRef	➔ Rango de celdas a buscar
intervaloRetorno	CellRef	➔ Rango de celdas para encontrar el valor correspondiente

Descripción

La función **LOOKUP** busca el valor *aBuscar* en el rango de celda *intervaloABuscar* y devuelve el valor correspondiente utilizado en el rango *intervaloRetorno*.

aBuscar debe contener la referencia de una celda que en realidad contiene el valor a encontrar.

intervaloABuscar debe ordenarse en orden ascendente, ya que **LOOKUP** utiliza el primer valor superior o igual al valor establecido como *aBuscar*.

Si no se encuentra *aBuscar*, coincide con el valor más grande en *intervaloABuscar* que es menor o igual a *aBuscar*.

Ejemplo

	A	B	C
1		1	520
2		2	380
3		3	697
4		4	437
5		5	578
6		6	185
7			
8		3	
9	=LOOKUP(A8,B1:B6,C1:C6)		
10			

"3", ubicado en la celda A8, es el valor que se debe buscar. B1: B6 es el intervalo de búsqueda. C1: C6 es el intervalo de retorno. La celda B3 contiene el valor a encontrar. El valor correspondiente en el intervalo de retorno está en la celda C3, es decir, "697".

MAX

MAX (valor {, valor2 , ... , valorN})

Parámetro	Tipo	Descripción
valor	Real, Array	→ Número o array numérico a evaluar

Descripción

La función **MAX** devuelve el valor mayor (máximo) de *valor*. Acepta y devuelve datos numéricos. En *valor*, pase los valores para evaluar. Cada argumento puede ser un número o un array de números. Puede usar un solo array (rango de celdas) o varios arrays (rangos de celdas). Si un array o referencia contiene texto, valores lógicos o celdas vacías, **MAX** ignora esos valores; sin embargo, las celdas con el valor cero se incluyen en los cálculos.

Ejemplo

```
MAX(A1,B2,C3,D4,E5)
```

```
MAX(A1:A9)
```

```
MAX(2,15,12,3,7,19,4) //19
```

MID (valor , iniciaDesde {, numCaracteres})

Parámetro	Tipo	Descripción
valor	Texto, CellRef, Expresión	→ Texto que contiene los caracteres a extraer
iniciaDesde	Número	→ Número que designa al primer caracter a extraer en el texto
numCaracteres	Número	→ Número de caracteres a devolver

Descripción

La función **MID** devuelve el número solicitado de caracteres desde el valor que comienza en la posición especificada en *iniciaDesde*. Acepta datos de texto para *valor* y datos numéricos para *iniciaDesde* y *numCaracteres*. Devuelve datos de texto.

En *valor*, pase la cadena de texto que contiene los caracteres que desea extraer. Puede ser una cadena, una fórmula que devuelve una cadena o una referencia a una celda que contiene una cadena.

En *iniciaDesde*, pase un número que represente el primer carácter que desee extraer en el texto, con el primer carácter en el texto que tenga un valor de uno (1); si no es un número entero, el número se trunca. Puede ser una cadena, una fórmula que devuelve una cadena o una referencia a una celda que contenga una cadena. Si *iniciaDesde*:

- es mayor que la longitud de *valor*, se devuelve una cadena vacía (" ")
- es menor que la longitud de *valor*, pero *iniciaDesde*+ *numCaracteres* excede la longitud del *valor*, se devuelven los caracteres hasta el final del texto.

En *numChars*, pase el número de caracteres a devolver de *valor*; si no se especifica un número entero, el número se trunca.

Ejemplo

```
MID(B17,5,8)
```

```
MID("hello world",7,20) //"world"
```

MIN

MIN (valor {, valor2 , ... , valorN})

Parámetro	Tipo	Descripción
valor	Real, Array	→ Número o array numérico a evaluar

Descripción

La función **MIN** devuelve el valor de valor más bajo (mínimo) de *valor*. Acepta y devuelve datos numéricos.

En *valor*, pase los valores a evaluar. Cada argumento puede ser un número o un array de números (un array (rango de celdas) o varios arrays (rangos de celdas)).

Si un array o referencia contiene texto, valores lógicos o celdas vacías, **MIN** ignora esos valores; sin embargo, las celdas con el valor cero se incluyen en los cálculos.

Ejemplo

```
MIN(2,15,12,3,7,19,4) //2
```

NOT

NOT (valorLogico)

Parámetro	Tipo		Descripción
valorLogico	Booleano, Número, Expresión	→	Valor a evaluar

Descripción

La función **NOT** invierte el valor lógico de su parámetro.

valorLogico puede ser un booleano o un número. Si su valor es cero, la función devuelve TRUE. Si es un valor distinto de cero, la función devuelve FALSE.

Ejemplo

```
NOT(A3)
```

```
NOT(D5>100)
```

```
NOT(0) //TRUE
```

```
NOT(TRUE) //FALSE
```

```
NOT(12) //FALSE
```

NOW

NOW ()

Este comando no requiere parámetros

Descripción

La función **NOW** devuelve la hora de la fecha actual. Devuelve un objeto DateTime.

Esta función se actualiza cuando se vuelve a calcular la hoja de cálculo o la celda que contiene la función.

Ejemplo

Si hoy es lunes 8 de enero de 2018 a las 11:25:42:

```
NOW() //1/8/2018 11:25:42
```

NPER

NPER (i , m , p {, f})

Parámetro	Tipo	Descripción
i	Número	→ Tasa de interés para un periodo
m	Número	→ El pago mensual pagado al final del período (use un valor negativo). Para el valor adquirido: pase un parámetro vacío (ver ejemplo)
p	Número	→ El valor actual del préstamo
f	Número	→ El valor futuro del préstamo (use un valor negativo)

Descripción

La función **NPER** devuelve la cantidad de períodos necesarios para reembolsar un préstamo. Se pueden usar dos fórmulas:

- Primera fórmula, cuando conoces el pago mensual:

$$\text{NPER}(i,m,p) = \frac{\log((m - 1 \times p)/m)}{\log(1 + i)}$$

- Segunda fórmula, cuando conoces el pago total:

$$\text{NPER}(i,,f,p) = \frac{\log(f/p)}{\log(1 + i)}$$

Ejemplo 1

Usted pidió prestado 6.500 euros con 10,5% de interés anual y reembolsa 166,42 euros por mes:

```
NPER(10.5%/12,-166.42,6500) //48
```

Ejemplo 2

Pidió prestado 3.000 euros con un interés anual del 10,5% y sabe que el total de los pagos mensuales será de 5.000 euros:

```
NPER(10.5%/12,,3000,-5000) //58
```

OR

OR (valorLogico {, valorLogico2 , ... , valorLogicoN})

Parámetro	Tipo	Descripción
valorLogico	Booleano, Array, Expresión	→ Valor(es) a evaluar

Descripción

La función **OR** devuelve False si la evaluación de todos los parámetros es falsa. Devuelve True si la evaluación de al menos uno de los parámetros es verdadera.

La función acepta valores booleanos como numéricos (0 o 1) o expresiones lógicas (TRUE o FALSE) para hasta 255 argumentos. También puede especificar un único array en lugar de listar los valores por separado, o hasta 255 arrays. También puede especificar el valorLogico como una expresión.

Ejemplo

```
OR(B3,B6,B9)
```

```
OR(D2:D12)
```

```
OR(TRUE,FALSE,FALSE) //TRUE
```

```
OR(TRUE()) //TRUE
```

```
OR(FALSE(),FALSE()) //FALSE
```

```
OR(1+1=1,2+2=5) //FALSE
```

```
OR(5+3=8,5+4=12) //TRUE
```

PMT

PMT (i , n , p)

Parámetro	Tipo		Descripción
i	Número	→	La tasa de interés por un período
n	Número	→	El número de períodos
p	Número	→	El valor actual del préstamo (use un valor negativo)

Descripción

La función **PMT** devuelve el valor de los pagos mensuales del préstamo.

Aquí está la fórmula para **PMT**:

$$\text{PMT}(i,n,p) = \frac{i}{1 - (1+i)^{-n}}$$

Ejemplo

Usted pidió prestado 6.500 euros a 48 meses con un 10,5% de interés:

```
PMT(10.5%/12,48,-6500) //166.42
```


PV (i , n , m {, f})

Parámetro	Tipo	Descripción
i	Número	→ La tasa de interés por un período
n	Número	→ Número de periodos
m	Número	→ Para interés compuesto: el pago mensual al final de cada período (use un valor negativo). Para interés individual: pase un parámetro vacío (ver el ejemplo)
f	Número	→ Para tasa de interés simple: el valor final al final de un período (use un valor negativo)

Descripción

La función **PV** calcula:

- el valor actual de una suma usando el interés compuesto, o
- el valor actual de una suma usando tasas de interés individuales.

Para calcular el valor actual de una suma usando el interés compuesto, pase el parámetro *m* y omita el parámetro *f*. Está es la fórmula para este cálculo:

$$PV(i,n,m) = m \times \frac{1 - (1 + i)^{-n}}{i}$$

Para calcular el valor actual de una suma usando tasas de interés simples, pase el parámetro *f* y pase un parámetro vacío (,,) para el marcador de posición *m*. Está es la fórmula para este cálculo:

$$PV(i,n,,f) = \frac{f}{(1 + i)^n}$$

Ejemplo 1

Interés compuesto: tiene un préstamo con una tasa de interés anual del 12% (por lo tanto, 1% por mes) durante 5 meses con un pago mensual de 1.000 €.

```
PV(1%,5,-1000) //4853,4312393251
```

Ejemplo 2

Tasa de interés simple: tiene un préstamo con una tasa de interés anual del 12% (por lo tanto, 1% por mes) durante 5 meses con un pago mensual de 1.000 €.

Nota: preste atención al doble " , , " dentro de la sintaxis.

```
PV(1%,5,, -5*1000) //4757,328438033744
```

RAND

RAND ()

Este comando no requiere parámetros

Descripción

La función **RAND** devuelve un número aleatorio entre 0 y 0,9999999...

Ejemplo

```
RAND()  
RAND()*100  
INT(RAND()*100)
```

⚙ RATE

RATE (n , m , p {, f})

Parámetro	Tipo	Descripción
n	Número	→ Número de períodos
m	Número	→ El pago mensual pagado al final del período (use un número negativo). Para el valor futuro: pase un parámetro vacío (ver ejemplo)
p	Número	→ El valor actual del préstamo
f	Número	→ El valor futuro (use un número negativo)

Descripción

La función **RATE** devuelve la tasa de interés correspondiente a los valores pasados en los parámetros. Se pueden usar dos fórmulas:

- Primera fórmula, cuando conoce el pago mensual pagado al final del período

$$i = \frac{m \times (1 - (1 + i)^{-n})}{p}$$

- Segunda fórmula, cuando conoces el valor adquirido:

$$u_{n+1} = \frac{m \times (1 - (1 + u_n)^{-n})}{p}$$

Ejemplo 1

Usted pidió prestado 3.000 euros y sus pagos mensuales son de 1.000 euros en 5 meses:

```
RATE(5,-1000,3000) //0.19
```

Ejemplo 2

Usted pidió prestado 2.800 euros y sus pagos mensuales son de 6.000 euros en 5 meses:

```
RATE(5,,2800,-6000) //0.16
```

ROUND

ROUND (valor , decimales)

Parámetro	Tipo		Descripción
valor	Número	→	Número a redondear
decimales	Número	→	Número de decimales

Descripción

La función **ROUND** redondea el *valor* especificado al número más cercano, utilizando el número de *decimales* especificado.

Use el parámetro *valor* para especificar el número a redondear. Puede pasar cualquier dato numérico (o referencia de celda que contenga datos numéricos).

Use el parámetro *decimales* para especificar la cantidad de decimales. Este parámetro tiene estas reglas:

- Defina el número de decimales para un valor > 0 para redondear al número especificado de lugares decimales.
- Defina el número de decimales a cero para redondear al número entero más cercano.
- Defina el número de decimales a un valor < 0 para redondear el valor que queda del decimal al diez, ciento, etc. más cercanos.

Ejemplo

```
ROUND(A3,-2)
```

```
ROUND(C4,B2)
```

```
ROUND(PI(),5) //3.14159
```

```
ROUND(29.2,-2) //0 porque 29.2 es más cercano a 0 que a 100.
```

```
ROUND(-1.963,0) //-2
```

ROW

ROW ({referencia})

Parámetro	Tipo	Descripción
referencia	CellRef	Una celda o un rango de celdas

Descripción

La función **ROW** devuelve el número de fila de *referencia*.

referencia puede ser una celda o un rango de celdas. Si se omite el argumento de *referencia*, el argumento predeterminado es la referencia de la celda en la que se coloca la función **ROW**.

Ejemplo

```
ROW(B2) // 2
```

```
ROW(B1:B5) //1
```

SIN

SIN (valor)

Parámetro	Tipo	Descripción
valor	Real	Ángulo cuyo seno se devuelve

Descripción

La función **SIN** devuelve el seno del ángulo especificado en *valor*. Acepta y devuelve datos numéricos. En *valor*, pase todo número real (un ángulo) para el cual devolver el seno. Debe expresarse en radianes. Si el ángulo está en grados, multiplíquelo por $\text{PI}/180$ para convertirlo a radianes.

Ejemplo

```
SIN(B4)
```

```
SIN(30*PI()/180) //0.5
```

SQRT

SQRT (valor)

Parámetro	Tipo	Descripción
valor	Número	→ Número ≥ 0 para obtener la raíz cuadrada

Descripción

La función **SQRT** devuelve la raíz cuadrada positiva del *valor* especificado. *valor* debe ser un número positivo (o 0); de lo contrario, se devuelve un error.

Ejemplo

```
SQRT(B4)
```

```
SQRT(256) //16
```

STDEV.P

STDEV.P (valor {, valor2 , ... , valorN})

Parámetro	Tipo	Descripción
valor	Número, CellRef	→ Valores a evaluar

Descripción

La función **STDEV.P** devuelve la desviación estándar para toda una población de valores numéricos. La desviación estándar es una medida de la dispersión de valores en relación con el promedio (valor promedio).

En *valor*, *valor2...*, pase los argumentos numéricos correspondientes a la población. Cada argumento puede ser una celda, un rango de celdas o un número. Esta función puede tener hasta 255 argumentos.

Nota: si sus datos representan una muestra de la población, calcule la desviación estándar utilizando la función [STDEV](#).

STDEV.P se calcula utilizando el método "biased" o "n". La función **STDEV.P** utiliza la siguiente fórmula:

$$\sqrt{\frac{\sum (x - \bar{x})^2}{n}}$$

Ejemplo

STDEV.P(A1,B2,C3,D4,E5,F6)

STDEV.P(A1:A9)

STDEV.P(95,89,73,87,85,76,100,96,96) da el resultado 8.8079649700473

SUBSTITUTE

SUBSTITUTE (*str* , *aReemplazar* , *reemplazo* {, *instance*})

Parámetro	Tipo	Descripción
<i>str</i>	CellRef	→ Cadena o referencia a una celda que contiene la cadena en la que desea reemplazar los caracteres
<i>aReemplazar</i>	Texto	→ Cadena a reemplazar
<i>reemplazo</i>	Texto	→ Nueva cadena a usar en lugar de cadena existente
<i>instance</i>	Número	→ Qué ocurrencia de la cadena existente reemplazar; de lo contrario, toda ocurrencia es reemplazada

Descripción

La función **SUBSTITUTE** reemplaza a *aReemplazar* en el texto *str* con *reemplazo* y devuelve el texto editado.

Por defecto, **SUBSTITUTE** reemplaza la primera ocurrencia de *aReemplazar*. Pase *instancia* para definir qué ocurrencia desea reemplazar.

Si no se encuentra *aReemplazar*, **SUBSTITUTE** devuelve *str*.

Ejemplo

```
SUBSTITUTE("Hello You","You","World") //Hello World
```

SUM

SUM (valor {, valor2 , ... , valorN})

Parámetro	Tipo	Descripción
valor	Número, CellRef	Número(s) a adicionar

Descripción

La función **SUM** devuelve la suma de celdas o rango de celdas.

En *valor*, pase los valores a evaluar. Cada argumento puede ser un número o un array de números (un único array (rango de celdas) o varios arrays (rangos de celdas)).

Nota: el operador + ofrece una auto conversión para valores no numéricos pasados por constante y para valores no numéricos pasados por referencia. La función **SUM** ofrece una auto conversión para valores no numéricos pasados por constante pero ignora los valores no numéricos pasados por referencia.

Ejemplo

```
SUM(A1,B7,C11)
```

```
SUM(A1:A9)
```

```
SUM(A2:A14,B2:B18,D12:D30)
```

```
SUM(95,89,73,87,85,76,100,96,96) //797
```

TAN

TAN (valor)

Parámetro	Tipo	Descripción
valor	Real	→ Ángulo cuya tangente se devuelve

Descripción

La función **TAN** devuelve la tangente del ángulo especificado en *valor*. Acepta y devuelve datos numéricos. En *valor*, pase todo número real (un ángulo) por el cual devolver la tangente. Debe expresarse en radianes. Si el ángulo está en grados, multiplíquelo por $\text{PI}/180$ para convertirlo a radianes.

Ejemplo

```
TAN(B3)
```

```
TAN(45*PI()/180) //1
```

TEXT

TEXT (valor , formato)

Parámetro	Tipo		Descripción
valor	Número, CellRef	→	Valor numérico a formatear en texto
formato	Texto	→	Formato a aplicar al valor

Descripción

La función **TEXT** devuelve una cadena compuesta de un número *valor* formateado según el *formato*. Pase un dato numérico o una referencia a una celda que contenga datos numéricos en *valor* y un argumento de texto en *formato*.

Ejemplo

```
TEXT(A1,"$0.00") // $10.00 si A1 contiene 10
```

```
TEXT(100,"0.00€") // 100.00€
```

TODAY ()

Este comando no requiere parámetros

Descripción

La función **TODAY** devuelve la fecha y la hora de la fecha actual. Devuelve un objeto DateTime. Esta función se actualiza cuando se vuelve a calcular la hoja de cálculo o la celda que contiene la función.

Ejemplo

Si el día actual es el lunes 8 de enero de 2018:

```
TODAY() //1/8/2018 0:00:00
```

TRUE

TRUE ()

Este comando no requiere parámetros

Descripción

La función **TRUE** devuelve el valor lógico TRUE (1).

Ejemplo

```
TRUE() //TRUE
```

TYPE

TYPE (valor)

Parámetro

valor

Tipo



Descripción

Valor a evaluar

Descripción

La función **TYPE** devuelve el tipo de *valor* como un número.

Los tipos devueltos son:

Tipo de valor	Número devuelto
Número	1
Objeto DateTime	1
Objeto TimeSpan	1
Texto	2
Booleano	4
Error	16
Array	64

Utilice la función **TYPE** cuando la ejecución de otra función dependa del tipo de valor contenido en una celda específica. La función **TYPE** es particularmente útil cuando se llaman funciones que aceptan diferentes tipos de datos.

Ejemplo

```
TYPE(G15)
```

```
TYPE(42) //1
```

```
TYPE("String") //2
```

```
TYPE(TRUE) //4
```

VAR.P

VAR.P (valor {, valor2 , ... , valorN})

Parámetro	Tipo	Descripción
valor	Número, Array	→ Valores para obtener la varianza

Descripción

La función **VAR.P** devuelve la varianza en función de la población completa, que utiliza valores numéricos. En *valor*, pase los valores a evaluar. Cada argumento puede ser un número o un array de números (un único array (rango de celdas) o varios arrays (rangos de celdas)).

Nota: esta función asume que sus argumentos son la población completa. Si sus datos representan solo una muestra de la población, calcule la varianza usando la función [VAR.S](#).










Ejemplo

```
VAR.P(B3,C4,B2,D10,E5)
```

```
VAR.P(A1:A9)
```

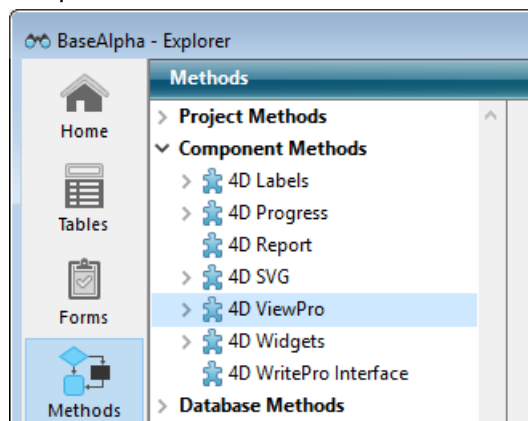
```
VAR.P(98,85,76,87,92,89,90) //39.265306122449
```


Lenguaje 4D View Pro

-  Sobre los comandos 4D View Pro
-  Gestión de áreas 4D View Pro
-  Evento formulario On VP Ready
-  VP Export to object
-  VP Convert from 4D View
-  VP EXPORT DOCUMENT
-  VP IMPORT DOCUMENT
-  VP IMPORT FROM OBJECT
-  VP NEW DOCUMENT

🧩 Sobre los comandos 4D View Pro

La funcionalidad de hoja de cálculo 4D View Pro es un componente 4D integrado. Por lo tanto, el elemento **4D ViewPro** aparece en la página Métodos del Explorador de la base, en la sección "Métodos de componente":



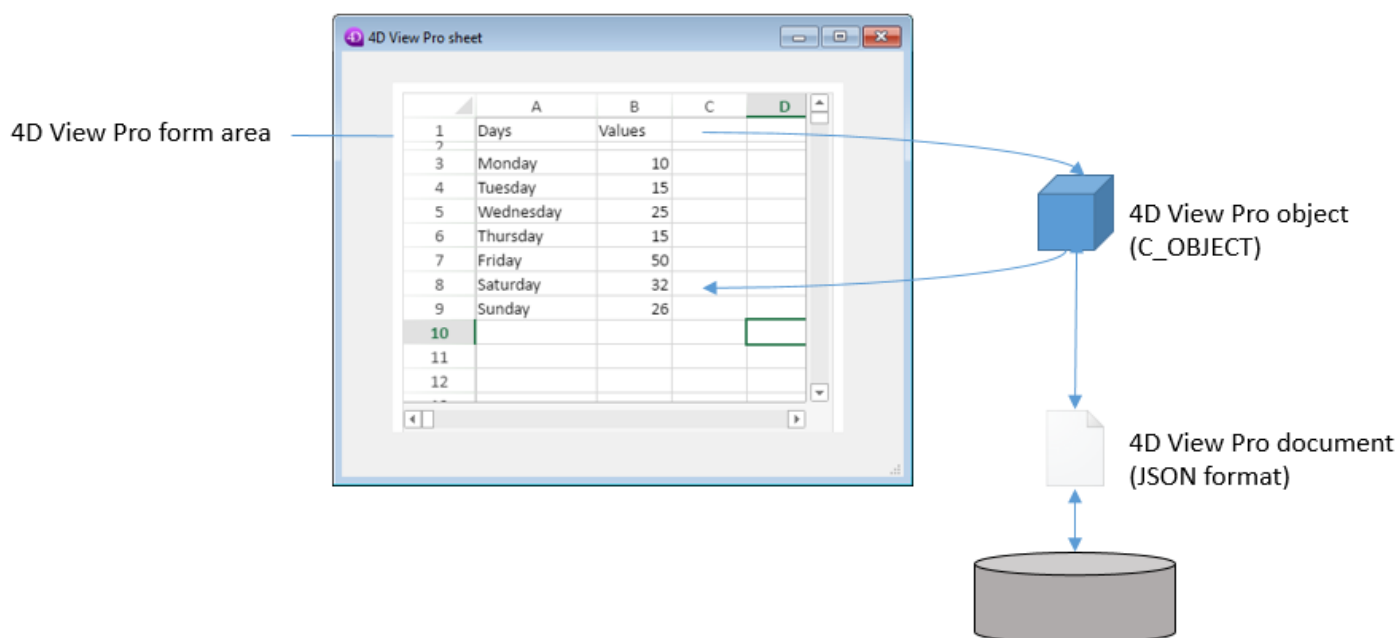
Puede expandir este elemento para ver todos los comandos del componente. Estos comandos se pueden utilizar en el editor de métodos 4D como los comandos de lenguaje 4D.

🌱 Gestión de áreas 4D View Pro

Arquitectura

Al trabajar con las áreas de 4D View Pro en sus formularios, necesita manejar varios elementos:

- El **área de formulario 4D View Pro** (objeto de formulario 4D): contiene y muestra el objeto 4D View Pro. Esta área se define por un nombre (campo "Nombre del objeto" en la lista de propiedades).
- El **objeto 4D View Pro** (variable o expresión de tipo **C_OBJECT**): almacena todo el contenido de la hoja de cálculo (ver abajo). Se asocia un objeto 4D View Pro a un área de formulario 4D View Pro a través del campo "Variable o expresión" de la lista de propiedades.
- El **documento 4D View Pro** (documentos .4vp): almacena todo el contenido de la hoja de cálculo en formato JSON.



Al cargar un objeto 4D View Pro en un área de formulario, 4D genera el evento de formulario On VP Ready una vez se ha cargado toda el área. Debe ejecutar cualquier código 4D View Pro que maneje el área en este evento, de lo contrario se devuelve un error.

Objeto 4D View Pro

El objeto 4D View Pro describe el documento y es manejado automáticamente por los comandos de 4D View Pro. Contiene las siguientes propiedades:

Propiedad	Tipo de valor	Descripción
version	Entero largo	Versión del componente interno
dateCreation	Timestamp	Fecha de creación
dateModified	Timestamp	Fecha de la última modificación
meta	Objeto	Contenido gratuito, reservado para el desarrollador 4D
spreadJS	Objeto	Reservado para el componente 4D View Pro

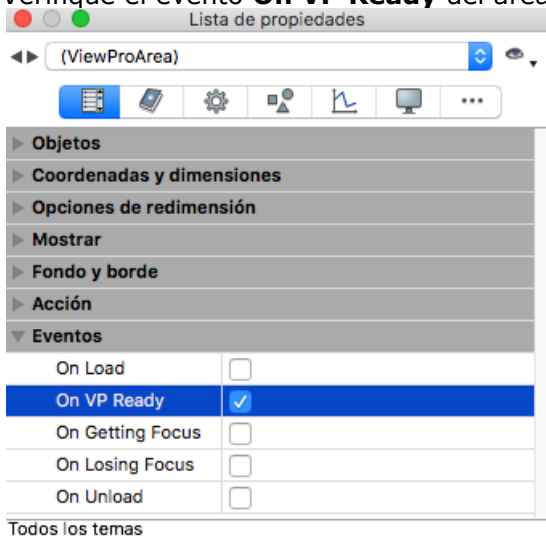
Evento formulario On VP Ready

Todo código de inicialización de área 4D View Pro, para cargar o leer valores desde o en el área, debe estar ubicado en el evento de formulario On VP Ready del área. Este **Evento formulario** se activa una vez que se ha completado la carga del área. Probar este evento asegura que el código se ejecutará en un contexto válido. Se devuelve un error si se llama a un comando 4D View Pro antes de generar el evento de formulario On VP Ready.

Nota: las áreas 4D View Pro se cargan asíncronamente en formularios 4D. Esto significa que el evento de formulario On load estándar no puede ser utilizado por el código de inicialización 4D View Pro, ya que podría ejecutarse antes de que se complete la carga del área. On VP Ready siempre se genera después de On load.

Para insertar código de inicialización para un área 4D View Pro:

1. Verifique el evento **On VP Ready** del área 4D View Pro en la lista de propiedades:



2. Escriba el código de inicialización del área 4D View Pro en el evento On VP Ready:

```
//método de formulario o método objeto de área  
Case of  
:(Form event=On VP Ready)  
... //código de inicialización  
End case
```

VP Export to object

VP Export to object (vpAreaName) -> Resultado

Parámetro	Tipo		Descripción
vpAreaName	Texto	→	Nombre del área 4D View Pro en el formulario
Resultado	Objeto	↩	Objeto 4D View Pro

Descripción

El comando **VP Export to object** devuelve el objeto 4D View Pro adjunto al área 4d View Pro *vpAreaName*. Puede utilizar este comando, por ejemplo, para guardar el área 4D View Pro en un campo objeto de la base de datos 4D.

Pase en *vpAreaName* el nombre del área en el formulario 4D. Si pasa un nombre no válido, se devuelve un error.

Para más información sobre los objetos 4D View Pro, consulte el párrafo [Objeto 4D View Pro](#) .

Ejemplo

Usted desea obtener la propiedad "versión" del área 4D View Pro actual:

```
C_OBJECT($vpAreaObj)
C_LONGINT($vpVersion)
$vpAreaObj:=VP Export to object("vpArea")
// $vpVersion:=OB Get($vpAreaObj;"version")
$vpVersion:=$vpAreaObj.version
```

VP Convert from 4D View

VP Convert from 4D View (documento4DView) -> Resultado

Parámetro	Tipo		Descripción
documento4DView	BLOB	→	Documento 4D View
Resultado	Objeto	↩	Objeto 4D View Pro

Descripción

El comando **VP Convert from 4D View** le permite convertir un documento 4D View heredado en un objeto 4D View Pro.

Nota: este comando no requiere que se instale el plug-in de 4D View heredado en su entorno.

En el parámetro *documento4DView*, pase una variable BLOB o campo que contiene el documento 4D View a convertir. El comando devuelve un *objeto* en el que toda la información almacenada originalmente en el documento 4D View se convierte en atributos de 4D View Pro.

La mayor parte de la información del documento 4D View se convierte directamente. Las propiedades o propiedades no soportadas que deben modificarse durante la conversión se listan en la página **Conversión de documentos 4D View**.

Ejemplo

Usted desea obtener un objeto 4D View Pro desde un área de 4D View almacenada en un BLOB:

```
C_OBJECT($vpObj)
$vpObj:=VP Convert from 4D View($pvblob)
```

VP EXPORT DOCUMENT

VP EXPORT DOCUMENT (vpAreaName ; rutaArchivo)

Parámetro	Tipo		Descripción
vpAreaName	Texto	→	Nombre del área 4D View Pro en el formulario
rutaArchivo	Texto	→	Ruta de acceso del documento

Descripción

El comando **VP EXPORT DOCUMENT** exporta el objeto 4D View Pro adjunto al área 4D View Pro *vpAreaName* a un documento en disco de acuerdo con el parámetro *rutaArchivo*.

Pase en *vpAreaName* el nombre del área en el formulario 4D. Si pasa un nombre no válido, se devuelve un error.

En *rutaArchivo*, pase la ruta de destino y el nombre del documento a exportar. Si sólo pasa el nombre del documento, se guardará en el mismo nivel que el archivo de estructura 4D.

La extensión del documento 4D View Pro es ".4vp". Se agrega automáticamente al nombre del documento si no se pasa en *rutaArchivo*.

Ejemplo

Usted desea exportar el contenido del área "VPArea" a un documento 4D View Pro en el disco:

```
C_TEXT($docPath)
```

```
$docPath:="C:\\Bases\\ViewProDocs\\MyExport.4VP"
```

```
VP_EXPORT_DOCUMENT("VPArea";$docPath)
```

```
//MyExport.4VP es guardado en el disco
```

VP IMPORT DOCUMENT

VP IMPORT DOCUMENT (*vpAreaName* ; *rutaArchivo*)

Parámetro	Tipo		Descripción
<i>vpAreaName</i>	Texto	→	Nombre del área 4D View Pro en el formulario
<i>rutaArchivo</i>	Texto	→	Ruta de acceso del documento

Descripción

El comando **VP IMPORT DOCUMENT** importa el documento 4D View Pro designado por *rutaArchivo* y lo muestra en el área 4D View Pro *vpAreaName*. El documento importado reemplaza todos los datos insertados en el área.

Pase en *vpAreaName* el nombre del área en el formulario 4D. Si pasa un nombre no válido, se devuelve un error.

En *rutaArchivo*, pase la ruta de destino y el nombre del documento importar. Sólo los documentos 4D View Pro (extensión ".4vp") son soportados por el comando. Debe pasar una ruta completa, a menos que el documento se encuentre al mismo nivel que el archivo de estructura de la base de datos, en cuyo caso puede pasar su nombre.

Se devuelve un error si el parámetro *rutaArchivo*, no es válido o si el archivo falta o está mal formado.

Ejemplo

Usted quiere importar un documento 4D View Pro almacenado en el disco, al abrir el formulario:

```
C_TEXT($docPath)
if(Form event=On VP Ready) //4D View Pro area loaded and ready
    $docPath:="C:\\Bases\\ViewProDocs\\MyExport.4VP"
    VP_IMPORT_DOCUMENT("VPArea";$docPath)
End if
```


VP IMPORT FROM OBJECT

VP IMPORT FROM OBJECT (vpAreaName ; viewPro)

Parámetro	Tipo		Descripción
vpAreaName	Texto	→	Nombre del área 4D View Pro en el formulario
viewPro	Objeto	→	Objeto 4D View Pro

Descripción

El comando **VP IMPORT FROM OBJECT** importa el objeto 4D View Pro *viewPro* y lo muestra en el área *vpAreaName*. El contenido del objeto importado reemplaza todos los datos insertados en el área.

Pase en *vpAreaName* el nombre del área en el formulario 4D. Si pasa un nombre no válido, se devuelve un error.

En *viewPro*, pase un objeto 4D View Pro válido. Este objeto puede haber sido creado usando **VP Export to object** o manualmente. Para más información sobre los objetos 4D View Pro, consulte la sección **Objeto 4D View Pro** .

Se devuelve un error si el objeto *viewPro* no es válido.

Ejemplo

Usted quiere importar una hoja de cálculo que se guardó previamente en un campo de objeto:

```
QUERY([VPWorkBooks];[VPWorkBooks]ID=10)  
VP IMPORT FROM OBJECT("ViewProArea1";[VPWorkBooks]SPBook)
```

VP NEW DOCUMENT

VP NEW DOCUMENT (*vpAreaName*)

Parámetro	Tipo	Descripción
<i>vpAreaName</i>	Texto →	Nombre del área 4D View Pro en el formulario

Descripción

El comando **VP NEW DOCUMENT** carga y muestra un nuevo documento por defecto en el objeto *vpAreaName* del área 4D View Pro en el formulario. El nuevo documento vacío reemplaza todos los datos ya insertado en el área.

Pase en *vpAreaName* el nombre del área en el formulario 4D. Si pasa un nombre no válido, se devuelve un error.

Ejemplo

Usted quiere mostrar un documento vacío en el objeto formulario "myVPArea":

```
VP NEW DOCUMENT("myVPArea")
```