

# 4D View Pro

-  Présentation
-  Fonctionnalités de list box avancées
-  Création d'une zone 4D View Pro
-  Utilisation d'une zone 4D View Pro
-  Conversion des documents 4D View
-  Formules 4D View Pro
-  Langage 4D View Pro
-  Liste alphabétique des commandes

# ➤ Présentation

## A propos de 4D View Pro

---

4D View Pro fournit un ensemble de fonctionnalités relatives aux tableaux et aux présentations en liste. 4D View Pro est destiné à offrir aux utilisateurs 4D une alternative moderne et intégrée aux principales fonctionnalités du plug-in d'ancienne génération 4D View.

4D View Pro se compose de deux parties :

- un ensemble de fonctionnalités list box avancées offrant un contrôle complet des hauteurs de lignes,
- un composant associé à une zone de formulaire 4D permettant aux développeurs d'inclure une zone de tableur dans leurs formulaires (*préversion dans 4D v16 R4*).

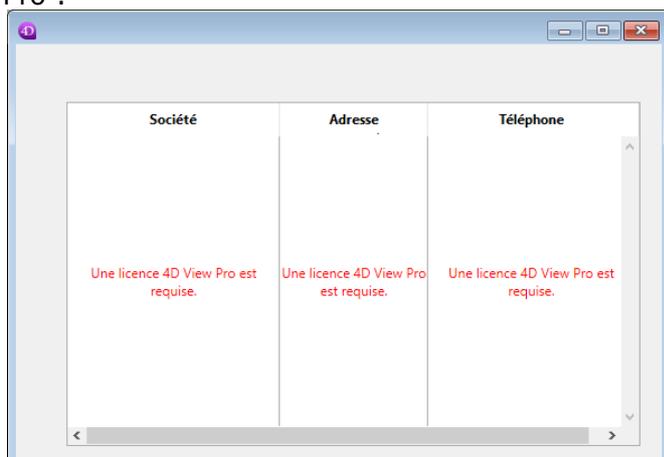
## Installation et activation

---

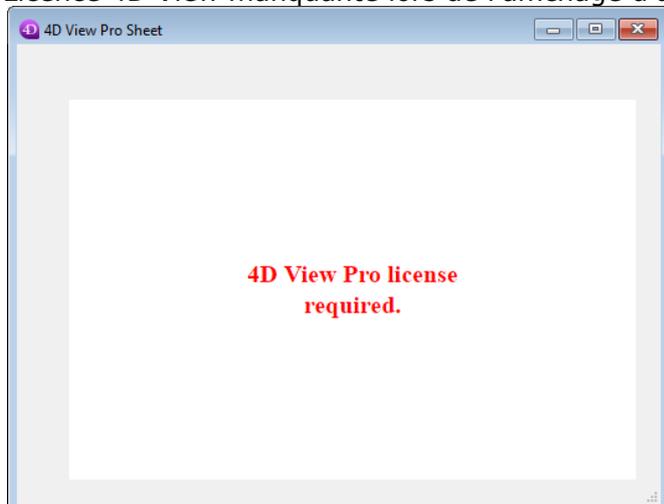
A la différence de l'ancien plug-in 4D View, 4D View Pro est directement inclus dans l'application 4D, facilitant son déploiement et la maintenance de ses fonctionnalités. Aucune installation supplémentaire n'est requise.

Cependant, 4D View Pro requiert la même licence que 4D View. La licence 4D View doit être installée dans votre application pour que vous puissiez utiliser les fonctionnalités de 4D View Pro. Lorsque cette licence n'est pas installée, le contenu d'un objet utilisant une fonctionnalité 4D View Pro (list box ou zone 4D View Pro) n'est pas affiché à l'exécution et un message d'erreur est affiché à sa place.

- Licence 4D View manquante lors de l'affichage d'une list box utilisant une fonctionnalité 4D View Pro :



- Licence 4D View manquante lors de l'affichage d'une zone de tableur 4D View Pro :





# Fonctionnalités de list box avancées

Comme décrit dans la section **Présentation**, 4D View Pro est constitué pour partie d'un ensemble de fonctionnalités de list box. Ces fonctions avancées sont :

- Association de tableaux objets aux colonnes de list box :

Label	Value
Document Name	MyReport
Document Type	PDF
Reference	123456
Category	<input type="text" value=""/>
Include Abstract	<input checked="" type="checkbox"/>
Printable area size (height)	297 <input type="text" value="mm"/>
Printable area size (width)	210 <input type="text" value="mm"/>
Show Preview	<input type="button" value="Preview..."/>

Pour plus d'informations, veuillez vous reporter à la page **Utiliser des tableaux objets dans les colonnes (4D View Pro)**.

- Contrôle des hauteurs de ligne dans les list box :

RowNum	Countries	Population
1	Luxembourg	502 202
2	Latvia	1 973 700
3	Kuwait	4 044 500
4	Croatia	4 284 889
5	Denmark	5 699 220
6	Nicaragua	6 071 045
7	Serbia	7 306 677
8	Honduras	8 249 574
9	Austria	8 572 895
10	Hungary	10 005 000
11	Czech Republic	10 674 947

Cette fonctionnalité est basée sur les éléments suivants :

- Propriété **Tableau hauteurs de lignes**.
- Commandes **LISTBOX Lire hauteur ligne** et **LISTBOX FIXER HAUTEUR LIGNE**.
- Constante **lk tableau hauteurs lignes** pour les commandes **LISTBOX Lire tableau** et **LISTBOX FIXER TABLEAU**.

Pour plus d'informations, veuillez vous reporter à la documentation de 4D.

- Hauteurs de lignes variables automatiques :

Picture	Name	Short text	Full text
	<b>ABORT</b>	The <b>ABORT</b> command is used from within an error-handling project method installed using the command <b>ON ERR CALL</b> .	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>excepting handler</i> ), and resumes the execution to the next line of code.
	<b>ASSERT</b>	The <b>ASSERT</b> 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 <b>ON ERR CALL</b> 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.
	<b>Asserted</b>	The <b>Asserted</b> command has an operation similar to that of the <b>ASSERT</b> command, with one difference in that it returns	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 <b>ASSERT</b> command.

Cette fonctionnalité est basée sur les éléments suivants :

- Propriété **Hauteur de ligne automatique** (disponible au niveau list box et colonne de list box)
  - Constante **lk hauteur ligne auto** pour les commandes **LISTBOX FIXER PROPRIETE** et **LISTBOX Lire propriété**
  - Commandes **LISTBOX FIXER HAUTEUR LIGNE AUTOMATIQUE** et **LISTBOX Lire hauteur ligne automatique** permettant de définir des limites pour les hauteurs de lignes.
- Pour plus d'informations, veuillez vous reporter à la documentation de 4D.



# Création d'une zone 4D View Pro

## Présentation

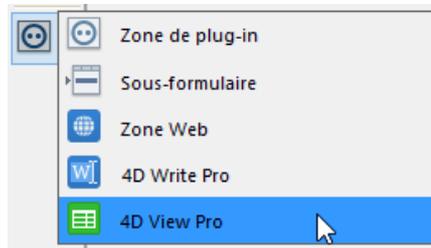
4D View Pro vous permet d'insérer et d'afficher une zone de tableur dans vos formulaires 4D. Une application tableur affiche généralement une grille de cellules dans lesquelles vous pouvez saisir des informations, réaliser des calculs ou afficher des images.

Après avoir créé des zones 4D View Pro dans vos formulaires, vous pouvez importer et exporter des feuilles de calcul à l'aide des commandes 4D View Pro.

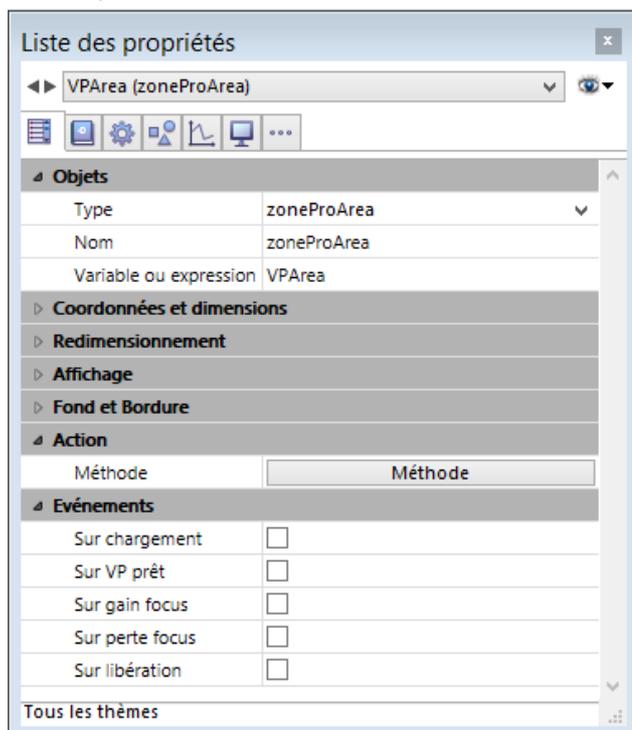
**Compatibilité :** Les zones 4D View Pro ne sont pas utilisables avec les versions 32 bits de 4D.

## Création de la zone

Les documents 4D View Pro sont affichés et modifiés via un objet de formulaire 4D nommé **4D View Pro**. Cet objet se trouve dans le dernier groupe de la barre d'objets de l'éditeur de formulaires de 4D (zone de plug-in, zone Web, etc.) :



Une zone formulaire 4D View Pro est configurée au moyen de propriétés standard présentes dans la liste des propriétés, telles que le **Nom** et **Variable ou Expression**, **Coordonnées**, **Affichage**, **Action**, et **Événements**.



- **Nom** : nom de la zone sur le formulaire 4D qui contient et affiche le document 4D View Pro.
- **Variable ou Expression** : nom de la variable d'objet de formulaire de la zone 4D View Pro.

Lorsque le formulaire est exécuté, la zone 4D View Pro affiche une zone de tableur par défaut :

	A	B	C	D	
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					

# Utilisation d'une zone 4D View Pro

## Présentation

---

Lorsqu'elles sont exécutées dans des formulaires, les zones 4D View Pro fournissent des fonctionnalités basiques de tableur comme l'édition de cellules et la saisie de formules. Les fonctionnalités plus avancées sont disponibles via le langage 4D View Pro.

## Principes de base de sélection, saisie et navigation

---

Les feuilles de calcul sont composées de lignes et de colonnes. Un nombre est associé à chaque ligne. Une lettre (ou un groupe de lettre si le nombre de colonnes dépasse le nombre de lettres de l'alphabet) est associée à chaque colonne. L'intersection d'une ligne et d'une colonne est une cellule. Les cellules peuvent être sélectionnées et leur contenu édité.

### Sélection

- Pour sélectionner une cellule, cliquez simplement sur celle-ci ou utilisez les touches Flèche sur votre clavier. Son contenu (ou formule) est affiché dans la cellule.
- Pour sélectionner plusieurs cellules contiguës, glissez la souris d'un bord à l'autre de la sélection. Vous pouvez aussi cliquer sur deux limites de la sélection en maintenant la touche **Maj** enfoncée.
- Pour sélectionner toutes les cellules de la feuille de calcul, cliquez sur la cellule  en haut à gauche de la zone.
- Pour sélectionner une colonne, cliquez sur la lettre correspondante (ou le groupe de lettres).
- Pour sélectionner une ligne, cliquez sur le nombre correspondant.
- Pour sélectionner un groupe de cellules qui ne sont pas contiguës, maintenez la touche **Ctrl** (Windows) ou **Commande** (Mac) et cliquez sur chaque cellule à sélectionner.
- Pour désélectionner les cellules, cliquez n'importe où sur la feuille de calcul.

### Saisie et navigation

Double-cliquer dans une cellule vous permet de passer en mode saisie dans la cellule concernée. Si la cellule n'est pas vide, le curseur d'insertion est placé après le contenu de la cellule.



Les données peuvent être directement saisies une fois qu'une cellule est sélectionnée, même si le curseur d'insertion n'est pas visible. La saisie remplace alors le contenu de la cellule.

La touche **Tab** valide la saisie de la cellule et sélectionne la cellule à sa droite. La combinaison des touches **Maj + Tab** valide la saisie de la cellule et sélectionne la cellule à sa gauche.

La touche **Retour charriot** valide la saisie de la cellule et sélectionne la cellule en-dessous. La combinaison des touches **Maj + Retour charriot** valide la saisie de la cellule et sélectionne la cellule au-dessus.

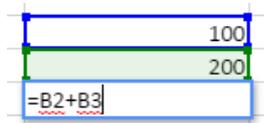
Les touches de direction (flèches) vous permettent de déplacer une cellule dans la direction indiquée par la flèche.

## Saisie d'une formule, d'une fonction ou d'une référence

---

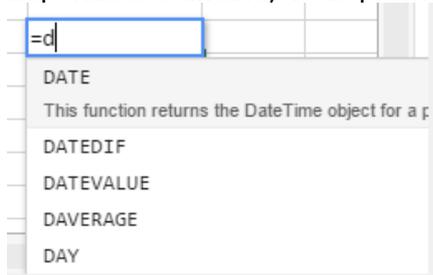
Pour saisir une formule ou une fonction dans une zone 4D View Pro :

1. Sélectionnez la cellule dans laquelle vous souhaitez saisir la formule ou la fonction.
2. Saisissez = (le signe égal).
3. Saisissez la formule  
OU  
Cliquez sur une cellule pour obtenir sa référence dans la formule.



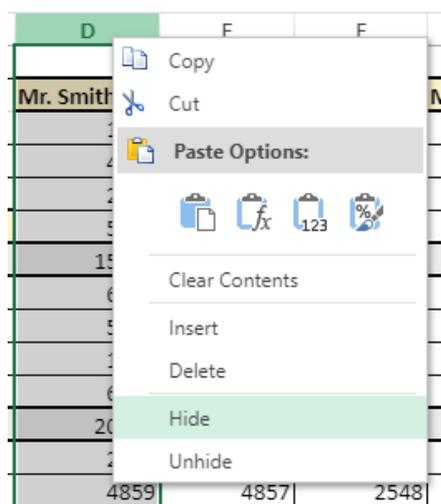
OU

Tapez la première lettre de la fonction à saisir. Un pop-up menu listant les fonctions et références disponibles s'affiche, vous permettant de sélectionner les éléments souhaités :



## Menu contextuel

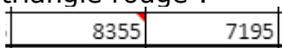
Les zones 4D View Pro bénéficient automatiquement d'un menu contextuel fournissant aux utilisateurs des fonctions d'édition standard telles que copier et coller, mais également des fonctions tableur basiques :



**Note :** Les commandes **Couper/Copier** et **Coller** du menu contextuel fonctionnent uniquement dans la zone de tableur, elles n'ont pas accès au presse-papiers système. En revanche, les raccourcis système tels que **Ctrl+c/Ctrl+v** sont utilisables et permettent l'échange de données entre la zone et les autres applications.

Ce menu propose des fonctionnalités supplémentaires selon la zone dans laquelle vous avez cliqué :

- clic dans un en-tête de colonne ou de ligne : Insert, Delete, Hide ou Unhide,
- clic dans une cellule ou une plage de cellules :
  - **Filter** : permet de masquer la ligne en fonction de filtres (voir [Filtering rows](#) dans la documentation de SpreadJS).
  - **Sort** : trie le contenu de la colonne.
  - **Insert Comment** : permet à l'utilisateur de saisir un commentaire pour une zone. Lorsqu'un commentaire a été associé à une zone, la cellule en haut à gauche de la zone affiche un petit triangle rouge :



## Conversion des documents 4D View

Vous pouvez convertir vos documents 4D View en zones 4D View Pro à l'aide de la commande **VP Convert from 4D View**. La plupart des propriétés et des informations stockées dans vos documents 4D View sont automatiquement converties, y compris les formats, styles, valeurs, formules, sélections, zoom, etc. De manière générale, le rendu des documents 4D View dans 4D View Pro sera quasiment identique à celui leur version originale dans les zones 4D View, comme dans l'exemple suivant :

*Document 4D View original :*

**NOTE D'HONORAIRES**  
Suivant contrat du 14/05/12  
payable le 13/06/12

PROJ\_12\_024 - Villa de M. et Mme BEAUICHEMIN  
Construction d'une maison individuelle

BARZAC  
85 rue Boulangers  
45100 LA SOURCE

Éléments de missions			Avancement	
	%	Montant HT	Taux	Montant HT
Avant Projet Détaillé	20,00 %	4 180,00 €	100,00 %	4 180,00 €
Dossier de Permis de Construire	23,00 %	4 807,00 €	100,00 %	4 807,00 €
Assistance à la passation des contrats de travail	15,00 %	3 135,00 €	100,00 %	3 135,00 €
Direction de l'exécution des contrats de Travail	34,00 %	7 106,00 €	46,15 %	3 279,72 €
Assistance aux Opérateurs de Réception	8,00 %	1 672,00 €		
<b>TOTAL</b>	<b>100,00 %</b>	<b>20 900,00 €</b>		<b>15 401,72 €</b>

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 encasements - Exonérée 0% - Révisable pour retard : 3 fois le taux d'intérêt légal en vigueur  
SIRET 400 958 879 00065 - Code APE 742A - N° d'identification intracommunautaire : FR 49 800 932 884

*Document converti dans une zone 4D View Pro*

**NOTE D'HONORAIRES**  
Suivant contrat du 14/05/12  
payable le 13/06/12

PROJ\_12\_024 - Villa de M. et Mme BEAUICHEMIN  
Construction d'une maison individuelle

BARZAC  
85 rue Boulangers  
45100 LA SOURCE

Éléments de missions			Avancement	
	%	Montant HT	Taux	Montant HT
Avant Projet Détaillé	20,00 %	4 180,00 €	100,00 %	4 180,00 €
Dossier de Permis de Construire	23,00 %	4 807,00 €	100,00 %	4 807,00 €
Assistance à la passation des contrats de travail	15,00 %	3 135,00 €	100,00 %	3 135,00 €
Direction de l'exécution des contrats de Travail	34,00 %	7 106,00 €	46,15 %	3 279,72 €
Assistance aux Opérateurs de Réception	8,00 %	1 672,00 €		
<b>TOTAL</b>	<b>100,00 %</b>	<b>20 900,00 €</b>		<b>15 401,72 €</b>

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 encasements - Exonérée 0% - Révisable pour retard : 3 fois le taux d'intérêt légal en vigueur  
SIRET 400 958 879 00065 - Code APE 742A - N° d'identification intracommunautaire : FR 49 800 932 884

Nous faisons le maximum pour que les documents convertis soient les plus proches possibles des originaux, mais quelques fonctionnalités ne peuvent pas être intégralement rendues. Les limitations actuelles sont listées dans le paragraphe ci-dessous.

## Procédure de conversion

**Note :** La conversion des documents 4D View est améliorée de façon continue, sur la base des retours des utilisateurs. Il est fortement recommandé de conserver une copie de vos documents ou BLOBs 4D View originaux, même après avoir les avoir convertis.

En fonction de l'état courant de votre document de plug-in 4D View, la procédure de conversion requiert les étapes suivantes :

1. Chargez votre document 4D View (.4pv) dans un BLOB.

**Note :** Si votre document 4D View est déjà stocké dans un champ BLOB field, passez directement à l'étape 2.

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

2. Appelez la commande avec le BLOB contenant le document 4D View :

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

3. Assignez l'objet résultant à une zone de formulaire 4D View Pro afin de visualiser le résultat :

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

## Précisions sur la conversion

---

Le tableau suivant fournit le statut courant de la conversion initiale des zones 4D View. A noter que cette liste sera mise à jour régulièrement, à mesure que la procédure de conversion sera améliorée.

Fonctionnalité	Statut des conversions	Commentaires
Attributs des documents	<p>Les attributs d'affichage des documents sont convertis: cellules sélectionnées zoom, affichage de la grille.</p> <p>Les informations des documents sont converties : version, titre, auteur, sujet, société, note, date de création et de modification.</p>	
Colonnes et lignes	<p>Toutes les lignes et colonnes définies sont converties avec leur taille originale.</p> <p>Les en-têtes de lignes et de colonnes sont convertis sans restrictions.</p>	
Encadrements	<p>Les encadrements sont convertis avec leur épaisseur et couleur. Les paramètres d'affichage de la grilles sont convertis.</p>	<p>Un seul style de bordure simple et un seul style de bordure double sont disponibles dans 4D View Pro. Toutes les bordures simples et toutes les bordures doubles sont converties dans le style unique correspondant.</p>
Séparateurs	<p>Actuellement non convertis</p>	
Styles & polices	<p>Les styles et les feuilles de style sont convertis. Les styles conditionnels ne sont pas pris en charge.</p> <p>Les styles de rotation (texte orienté) ne sont pas convertis actuellement</p>	<p>Les styles obsolètes (<i>i.e.</i> Ombré, Condensé, etc.) et les polices QuickDraw ne sont pas convertis.</p>
Formats & noms de cellules	<p>Les formats des cellules sont convertis aux formats disponibles ayant un rendu similaire. Les formats de tous les types de données sont pris en charge : texte, numérique, date et time, booléen, image. Les noms de cellule sont convertis.</p> <p>Les formats utilisateurs 4D (débutant par un " ") ne sont pas convertis actuellement</p> <p>Les contrôles créés avec PV FIXER CONTROLE CELLULE (bouton / bouton radio / case à cocher / liste déroulante / combo box) ne sont pas convertis actuellement</p>	<p>Les conversions de formats peuvent produire des résultats non conformes pour certains documents car les formats peuvent s'appuyer parfois sur des paramètres système (par exemple pour le séparateur décimal) et peuvent avoir été définis globalement ou individuellement.</p>
Images	<p>Les images sont prises en charge et converties avec quelques limitations</p> <p>Les images au format tronqués centrée / mosaïque ne sont pas converties actuellement</p>	<p>Le format d'image 4D View contenait de multiples codecs et est désormais obsolète. Les images converties conservent uniquement le codec le plus approprié pour le rendu html (svg, png, jpeg, gif) et sont sauvegardées en base64.</p> <p>Les images d'arrière plan ne sont pas répliquées sur chaque page (ce concept n'existe pas dans 4D View Pro).</p>
Impression	<p>Actuellement non pris en charge</p>	<p>Les options et les paramètres d'impression définis dans le document 4D View sont convertis.</p>
Liens dynamiques	<p>Les cellules ou colonnes liées à des champs ou des variables ne sont pas prises en charge actuellement</p>	
Formules	<p>Les formules sont converties mais les références aux commandes, méthodes, variables et champs 4D ne sont pas prises en charge actuellement</p>	

# Formules 4D View Pro

## Présentation

---

Les fonctions 4D View Pro sont utilisées dans les formules. Chaque formule 4D View Pro est une expression qui retourne une valeur. Toutes les expressions sont composées d'opérandes et d'opérateurs :

- Les **opérateurs** (voir [Opérateurs et valeurs](#))
- Les **opérandes** sont divisés en plusieurs catégories :
  - valeurs,
  - références à d'autres cellules (relatives, absolues, mixtes ou par nom),
  - variables, champs et fonctions 4D
  - fonctions 4D View Pro.

Pour saisir une formule :

1. Sélectionnez la cellule dans laquelle vous souhaitez saisir la formule ou la fonction.
2. Saisissez = (le signe égal).
3. Entrez la formule, puis appuyez sur la touche Retour chariot.

Un grand nombre de fonctions sont disponibles dans 4D View Pro. Cette section propose un sous-ensemble des fonctions les plus utilisées. Pour la liste complète des fonctions prises en charge par 4D View Pro, veuillez consulter la [documentation de Spreadsheets](#).

-  Opérateurs et valeurs
-  Références des cellules
-  Conversion des formules du plug-in 4D View
-  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
-  PI
-  PMT
-  PV

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

## 🔧 Opérateurs et valeurs

### Opérateurs par type de données

---

4D View Pro supporte cinq types de données. Pour chaque type de données, des valeurs littérales et des opérateurs spécifiques sont pris en charge.

Types de données	Valeurs	Opérateurs
Nombre	1.2	+ (addition)
	1.2 E3	- (soustraction)
	1.2E-3	* (multiplication)
	10.3x	/ (division)
		^ (exponentiation, mise en puissance)
Date	10/24/2017	% (pourcentage - divise le nombre placé avant l'opérateur par cent)
		+ (date + nombre de jours -> date)
		+ (date + heure -> date + heure du jour)
		- (date - nombre de jours -> date)
Heure	10:12:10	- (date - date -> nombre de jours entre les deux)
		Duration operators:
		+ (addition)
		- (soustraction)
Chaîne	'Sophie' ou "Sophie"	* (durée * nombre -> durée)
		/ (durée / nombre -> durée)
Booléen	TRUE ou FALSE	& (concaténation)
		-

### Opérateurs de comparaison

---

Les opérateurs ci-dessous peuvent être utilisés avec deux opérandes de même type :

Opérateurs	Comparaison
=	égal à
<>	différent de
>	plus grand que
<	moins grand que
>=	supérieur ou égal à
<=	inférieur ou égal à

### Priorité des opérateurs

---

Liste des opérateurs, du plus important au moins important :

<b>Opérateur</b>	<b>Description</b>
()	Parenthèses (pour le groupage)
-	Negation logique
+	Plus
%	Pourcent
^	Exposant
* et /	Multiplication et division
+ et -	Addition et soustraction
&	Concaténation
= > < >= <= <>	Comparaison

## **Priorité des opérandes dans les formules**

---

Lorsque deux ou plusieurs opérandes différents portent le même nom, 4D View Pro détermine le type de chaque élément selon l'ordre de priorité suivant :

<b>Priorité</b>	<b>Type d'élément</b>
1	Référence de cellule
2	Nom de cellule
3	Fonction 4D View Pro
4	Méthode projet
5	Commande 4D
6	Variable

## ➤ Références des cellules

---

Les formules se réfèrent souvent à d'autres cellules au moyen d'adresses de cellules. Vous pouvez copier ces formules dans d'autres cellules. Lorsque vous copiez ou déplacez ces cellules vers de nouveaux emplacements, chaque adresse de cellule dans cette formule change ou reste la même en fonction de la façon dont elle est tapée. Une référence qui change est appelée une référence relative et se réfère à une cellule distante à gauche / à droite et au-dessus / en-dessous de la cellule avec la formule. Une référence qui pointe toujours vers une cellule particulière s'appelle une référence absolue. Par exemple, la formule suivante, saisie dans la cellule C8, ajoute les valeurs des deux cellules au-dessus et affiche le résultat :

= C6 + C7

Cette formule se réfère aux cellules C6 et C7. Autrement dit, 4D View Pro récupère les valeurs dans les cellules indiquées et les utilise dans la formule.

Vous pouvez également créer une référence mixte qui pointe toujours vers une ligne ou une colonne fixe.

### Notation des références

---

Si vous utilisez uniquement les coordonnées, par exemple, C5, 4D View Pro interprète la référence comme relative. Vous pouvez rendre la référence absolue en mettant un signe dollar devant la lettre ou le nombre, comme dans \$C\$5.

Vous pouvez mélanger des références absolues et relatives en insérant un signe dollar devant la lettre ou le numéro seul, par exemple, \$C5 ou C\$5. Une référence mixte vous permet de spécifier la ligne ou la colonne comme absolue, tout en permettant à l'autre partie de l'adresse d'être en référence relative.

Un moyen pratique, rapide et précis de spécifier une référence absolue est de nommer la cellule et d'utiliser ce nom à la place de l'adresse de la cellule. Une référence à une cellule nommée est toujours absolue.

Le tableau ci-dessous montre l'effet des différentes notations :

Exemple	Type de référence	Description
C5	Relative	Référence à l'emplacement relatif de la cellule C5, en fonction de l'emplacement de la cellule dans laquelle la référence est utilisée la première fois.
\$C\$5	Absolue	La référence est absolue. Elle va toujours se référer à la cellule C5, peu importe où elle est utilisée.
\$C5	Mixte	La référence se fait toujours à la colonne C, mais la référence à la ligne est relative à l'emplacement de la cellule où la référence est utilisée la première fois.
C\$5	Mixte	La référence se fait toujours à la ligne 5, mais la référence à la colonne est relative à l'emplacement de la cellule où la référence est utilisée la première fois.
Nom de cellule	Absolue	La référence est absolue. Elle va toujours se référer à la cellule nommée, peu importe où la référence est utilisée.

## Conversion des formules du plug-in 4D View

### Présentation

---

Comme indiqué dans la page [Conversion des documents 4D View](#), la plupart des contenus et des propriétés des documents du plug-in 4D View peuvent être convertis dans les documents 4D View Pro. Les formules sont également converties. Cependant, les langages des formules de 4D View et 4D View Pro sont un peu différents. Trois cas de compatibilité peuvent être rencontrés :

- une fonctionnalité 4D View (opérateur, valeur, fonction) existe à l'identique dans 4D View Pro : dans ce cas, la conversion est transparente.
- une fonctionnalité 4D View est prise en charge dans 4D View Pro mais via un opérateur ou une fonction différent(e) : dans ce cas, une correspondance automatique est effectuée au moment de la conversion.
- une fonctionnalité 4D View est prise en charge partiellement ou n'est pas prise en charge : dans ce cas, il vous sera nécessaire d'adapter vos formules converties afin qu'elles fonctionnent de la manière souhaitée.

Les tableaux suivants listent les fonctionnalités 4D View et leur équivalent dans 4D View Pro.

### Opérateurs

---

	<b>4D View</b>	<b>4D View Pro</b>
<b>Opérateurs numériques</b>		
Addition	+	+
Soustraction	-	-
Multiplication	*	*
Division	/	/
Modulo	\	<a href="#">MOD</a>
Division entière	÷	<a href="#">TRUNC(a/b)</a>
Exposant	^	^
Pourcentage	%	%
<b>Opérateurs booléens</b>		
AND	&	<b>AND</b>
OR		<b>OR</b>
Not	~	<b>NOT</b>
<b>Opérateurs sur les chaînes</b>		
concaténation	+	&
destruction	-	<b>SUBSTITUTE</b> , ex : "Down Trend"->"Down" est remplacé par SUBSTITUTE("Down Trend","Down","")
position	\	<a href="#">FIND</a> (sensible à la casse) ou <a href="#">SEARCH</a> (non sensible à la casse)
<b>Opérateurs sur les dates</b>		
jours+jours->date	+	+
date+heure->date+heure du jour	+	+
date-jours->date	-	-
date-date->nombre de jours	-	-
<b>Opérateurs sur les durées</b>		
addition	+	+
soustratction	-	-
multiplication	*	*
division	/	/
<b>Opérateurs de comparaison</b>		
égalité	=	=
différence	#	<>
supérieur à	>	>
inférieur à	<	<
supérieur ou égal à	>=	>=
inférieur ou égal à	<=	<=

## Fonctions

---

4D View	4D View Pro	Commentaire
Abs	<b>ABS</b>	
AddToDate	<b>DATE</b>	AddToDate(date;années;mois;jours) est remplacé par <b>DATE</b> ( <b>YEAR</b> (date)+années, <b>MONTH</b> (date)+mois, <b>DAY</b> (date)+jours)
date+heure	<b>TIME</b>	<b>DATE</b> (date) + <b>TIME</b> (heure)
And	<b>AND</b>	
ArcCos	<b>ACOS</b>	
ArcSin	<b>ASIN</b>	
ArcTan	<b>ATAN</b>	
Area	-	n/a (plus de zone de plug-in)
Average	<b>AVERAGE</b>	
Cell	<b>INDIRECT</b>	
Column	<b>COLUMNLETTER</b>	<b>COLUMN</b> retourne un numéro (et non une lettre)
Cos	<b>COS</b>	
Count	<b>COUNTA</b>	
CurrentDate	<b>TODAY</b>	
CurrentTime	<b>NOW</b>	
CVCompound	<b>PV</b>	CVCompound(1%;5;1000) est remplacé par <b>PV</b> (1%,5,-1000)
CVSimple	<b>PV</b>	CVSimple(1%;5;5*1000) est remplacé par <b>PV</b> (1%,5,, -1000) -- notez les deux virgules consécutives
Empty	<b>ISBLANK</b>	
Eval4D	-	Actuellement non disponible
Exp	<b>EXP</b>	
False	<b>FALSE</b>	
Find	<b>LOOKUP</b>	
FindCell	<b>FINDCELL</b>	
FVCompound	<b>FV</b>	FVCompound(1%;35;35*1000) est remplacé par <b>FV</b> (1%,35,-1000)
FVSimple	<b>FV</b>	FVSimple(12%;35;35*1000) est remplacé par <b>FV</b> (12%,35,, -35*1000) -- notez les deux virgules consécutives
If	<b>IF</b>	
Length	<b>LEN</b>	
Log	<b>LN</b>	
Max	<b>MAX</b>	
Min	<b>MIN</b>	
MonthlyValue	<b>PMT</b>	MonthlyValue(10.5%/12,48,6500) est remplacé par <b>PMT</b> (10.5%/12,48,-6500)
Not	<b>NOT</b>	
Or	<b>OR</b>	
PeriodNumber1	<b>NPER</b>	PeriodNumber1(10.5%/12;166.42;6500) est remplacé par <b>NPER</b> (10.5%/12,-166.42,6500)
PeriodNumber2	<b>NPER</b>	PeriodNumber2(10.5%/12,5000,3000) est remplacé par <b>NPER</b> (10.5%/12,,3000,-5000) -- notez les deux virgules consécutives
Pi	<b>PI</b>	
Random	<b>RAND</b>	
Range	<b>INDIRECT</b>	Range("A1";"A3") est remplacé par <b>INDIRECT</b> ("A1:A3") -- notez le caractère deux-points entre A1 et A3
Rate1	<b>RATE</b>	Rate1(5;1000;3000) est remplacé par <b>RATE</b> (5,-1000,3000)
Rate2	<b>RATE</b>	Rate2(5,6000,2800) est remplacé par <b>RATE</b> (5,,2800,-6000) -- notez les deux virgules consécutives
Rounding	<b>ROUND</b>	
Row	<b>ROW</b>	
Sin	<b>SIN</b>	
SquareRoot	<b>SQRT</b>	
StdDeviation	<b>STDEV.P</b>	
String	<b>TEXT</b>	

SubString	<b>MID</b>
Sum	<b>SUM</b>
Tan	<b>TAN</b>
True	<b>TRUE</b>
Type	<b>TYPE</b>
Variance	<b>VAR.P</b>

Les types dans 4D View Pro sont différents de ceux de 4D View

## ABS

ABS ( valeur )

Paramètre	Type	Description
valeur	Réel, Expression	→ Nombre dont la valeur absolue est retournée

### Description

---

La fonction **ABS** calcule la valeur absolue de la *valeur* spécifiée. Si la *valeur* est négative, une valeur positive sera retournée. Elle accepte les données numériques sous forme de valeurs ou d'expressions et renvoie des données numériques.

### Exemple

---

```
ABS(-6) //résultat:= 6
```

```
ABS(16-26) //résultat:= 10
```

```
ABS(6) //résultat:= 6
```

## ACOS

ACOS ( valeur )

Paramètre	Type	Description
valeur	Réel	→ Angle dont l'arccosinus est retourné. Doit être compris entre -1 et +1. Angle whose arccosine is returned. Must be between -1 and +1.

### Description

---

La fonction **ACOS** calcule l'angle de l'arccosinus spécifié dans *valeur*. *valeur* doit être compris entre -1 et +1.

L'angle retourné est en radians entre 0 et PI . Pour convertir le résultat en degrés, multipliez-le par 180/PI.

### Exemple

---

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

## AND

AND ( valeurLogique {, valeurLogique2 , ... , valeurLogiqueN} )

Paramètre	Type	Description
valeurLogique	Booléen, Numérique, Expression	→ Valeur(s) à évaluer

### Description

---

La fonction **AND** retourne TRUE si tous les arguments sont vrais ; sinon elle retourne FALSE si au moins un argument est faux.

Elle accepte les valeurs booléennes sous forme d'expressions numériques (0 ou 1) ou d'expressions logiques (TRUE ou FALSE) jusqu'à 255 arguments. Vous pouvez aussi spécifier un tableau simple à la place d'une liste de valeurs séparées, jusqu'à 255 tableaux. Vous pouvez aussi spécifier la valeur *valeurLogique* en tant qu'expression.

### Exemple

---

```
AND(D12,E12) //True if D12 and E12 cell values are true or 1, False otherwise
```

```
AND(D2:D12) //True if D12 to D12 cell values are true or 1, False otherwise
```

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

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

## **ASIN**

ASIN ( value )

<b>Paramètre</b>	<b>Type</b>	<b>Description</b>
value	Réel →	Sine of angle. Must be between -1 and +1.

### **Description**

---

The **ASIN** function calculates the arcsine, the angle whose sine is specified in *value*. It accepts and returns numeric data.

In *value*, specify the sine of the angle. The sine must be a value between -1 and +1.

The angle is returned in radians between  $-\pi/2$  and  $\pi/2$ . To convert the result to degrees, multiply it by  $180/\pi$ .

### **Exemple**

---

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

## ATAN

ATAN ( value )

Paramètre	Type	Description
value	Réel	→ Tangent of an angle. Must be between -1 and +1.

### Description

---

The ATAN function calculates the arctangent, i.e. the angle whose tangent is specified in *value*. It accepts and returns numeric data.

In *value*, specify the tangent of the angle to return. It must be between -1 and +1.

The angle is returned in radians between  $-\pi/2$  and  $\pi/2$ . To convert the result to degrees, multiply the result by  $180/\pi$ .

### Exemple

---

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

## AVERAGE

AVERAGE ( valeur {, valeur2 , ... , valeurN} )

Paramètre	Type	Description
valeur	Réel, Tableau	→ Nombre(s) dont la moyenne doit être calculée

### Description

---

La fonction **AVERAGE** calcule la moyenne de *valeur*.

Dans *valeur*, vous pouvez passer :

- des valeurs de type réel ou entier long,
- une plage ou plusieurs plages de cellules (tableau).

Jusqu'à 255 arguments peuvent être inclus.

### Exemple

---

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

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

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

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

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

## COLUMNLETTER

COLUMNLETTER ( {référence} )

Paramètre	Type		Description
référence	RefCell	→	Une cellule ou une plage de cellules

### Description

---

La fonction **COLUMNLETTER** retourne la lettre de la colonne de *référence*.  
*référence* peut être une cellule ou une sélection de cellules. Si l'argument *référence* est omis, l'argument par défaut est la référence de la cellule dans laquelle la fonction **COLUMNLETTER** est placée.

### Exemple

---

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

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

```
COLUMNLETTER() //current column letter
```

## COS

COS ( valeur )

Paramètre	Type	Description
valeur	Réel	→ Angle dont le cosinus est retourné

### Description

---

La fonction **COS** renvoie le cosinus de l'angle spécifié dans *valeur*. Elle accepte et renvoie des données numériques.

Dans *valeur*, passez n'importe quel nombre réel (un angle) dont vous voulez récupérer le cosinus. L'angle doit être exprimé en radians. Si l'angle est en degrés, multipliez-le par  $\text{PI}/180$  pour le convertir en radians.

### Exemple

---

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

COUNTA ( valeur {, valeur2 , ... , valeurN} )

Paramètre	Type	Description
valeur	Réel, Tableau →	Cellules ou plage de cellules à compter

## Description

---

La fonction **COUNTA** compte le nombre de cellules spécifiées dans *valeur* qui ne sont pas vides (c'est-à-dire les cellules qui contiennent des nombres, du texte ou des valeurs logiques). Elle accepte les références de cellules et retourne une valeur numérique.

Dans *valeur*, vous pouvez passer jusqu'à 255 cellules séparées ou un simple tableau de valeurs.

## Exemple

---

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

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

## **EXP**

EXP ( value )

<b>Paramètre</b>	<b>Type</b>	<b>Description</b>
value	Numérique	→ Number to evaluate



---

### **Description**

The **EXP** function returns the natural log base (e = 2.71828...) raised to the power of the number specified in *value*. It accepts and returns numeric data.

This function is the inverse of **LN**, so EXP(LN(x)) results in x.

---

### **Exemple**

```
EXP(B3)
```

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

## **FALSE**

FALSE ()

Ne requiert pas de paramètre

### **Description**

---

La fonction **FALSE** retourne la valeur logique FALSE (0).

### **Exemple**

---

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

## FINDCELL

FINDCELL ( toFind , searchRange )

Paramètre	Type		Description
toFind	RefCell	→	Value to find
searchRange	RefCell	→	Cells to search

### **Description**

---

The **FINDCELL** function searches for the *toFind* value in the *searchRange* range of cells and returns the reference of the cell in which it was found. This reference cannot be displayed, but can be used by other 4D View Pro functions that accept a cell reference (*CellRef*) as parameter.

*toFind* must contain the reference of a cell that actually contains the value to find.

### **Exemple**

---

Assuming cell C3 contains 10:

```
FINDCELL(C3,A1:B9) //returns 10 if the value is actually found in the A1:B9 cell range, otherwise it returns an error.
```

## FV

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

Paramètre	Type	Description
i	Numérique →	The interest rate for a period
n	Numérique →	The number of periods
m	Numérique →	For compound interest: the monthly payment at the end of each period (use a negative value). For single interest: pass an empty parameter (see example)
f	Numérique →	For single interest rate: the final value at the end of a period (use a negative value)

### Description

---

The **FV** function calculates:

- the final value of a sum using compound interest, or
- the final value of a sum using single interest

To calculate the value acquired during an investment, if the monthly payments are paid at the end of the period, pass the *m* parameter and omit the *f* parameter. Here is the formula for this calculation:

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

To calculate the final value of a sum using single interest, pass the *f* parameter and pass an empty parameter (,,) for the *m* placeholder. Here is the formula for this calculation:

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

### Exemple 1

---

**Compound interest:** you plan on depositing €1,000 each month in a savings account, which earns you 12% annual interest, for 35 months.

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

### Exemple 2

---

**Single interest rates:** same type of scenario as above.

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

## IF

IF ( testLogique , valeurSiVrai , valeurSiFaux )

Paramètre	Type	Description
testLogique	Expression	→ Valeur ou expression à évaluer
valeurSiVrai	Expression	→ Valeur à retourner si le résultat du test est vrai
valeurSiFaux	Expression	→ Valeur à retourner si le résultat du test est faux ou égal à 0

### Description

---

La fonction **IF** effectue une comparaison et renvoie l'une des deux valeurs en fonction de cette comparaison. Elle accepte des données numériques (booléennes - *en fait, elle accepte des tests sur tout type de données* et renvoie n'importe quel type de données.

La valeur de *testLogique* est évaluée. It must be or evaluate to numeric data ???, où des valeurs différentes de zéro indiquent TRUE, et une valeur à zéro indique FALSE. Elle peut contenir un des opérateurs de comparaison : plus grand que (>), inférieur à (<), égal à (=), ou différent de (<>). Si *testLogique* est :

- pas zéro (TRUE), alors *valeurSiVrai* est retournée.
- zéro (FALSE), alors *valeurSiFaux* est retournée.

### Exemple

---

Vous souhaitez évaluer B1, donnant le montant des ventes.

```
IF(B1<200,"Declining result","Good result") //"Good result" est affiché si B1>200
```

## INDIRECT

INDIRECT ( cell | cellRange )

Paramètre	Type	Description
cell   cellRange	RefCell →	Reference to a cell, a name defined as a reference, or a text string reference to a cell or to a range

### Description

---

The **INDIRECT** function returns the contents of *cell*. The *cell* parameter (mandatory) can be any cell reference, including absolute reference such as \$A\$1 or a character string. Use **INDIRECT** when you want to change the reference to a cell within a formula without changing the formula itself.

The **INDIRECT** function can also return the internal reference of a range of cells (reference cannot be displayed but can be used by other 4D View Pro functions).

### Exemple

---

```
INDIRECT("A1") //returns the contents of cell A1
```

```
COLUMN(INDIRECT("A1:A3")) //column 1  
ROW(INDIRECT("A1:F1")) //row 1
```

## ISBLANK

ISBLANK ( value )

Paramètre	Type	Description
value	RefCell, Expression, Numérique, Texte	→ Value to evaluate

### Description

---

The **ISBLANK** function tests if the contents of a cell, a number, a text, or any expression, is empty. This function returns TRUE if the value refers to an empty cell or to no data.

**Note:** Cells containing an empty string ("" ) are considered as blank.

### Exemple

---

```
IF(ISBLANK(A1);"Error";0) //"Error" if cell A1 is empty
```

```
ISBLANK(B1)
```

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

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

## LEN

LEN ( value )

Paramètre	Type	Description
value	Texte, RefCell	→ Text whose character length to count

### Description

---

The **LEN** function returns the number of characters in the *value* string.

In *value*, pass the text whose length you want to find. It must be a string or a cell reference to a string value.

**Note:** Spaces count as characters

### Exemple

---

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

## LN

LN ( value )

Paramètre	Type	Description
value	Numérique	→ Number greater than 0 to evaluate

### Description

---

The **LN** function returns the natural logarithm of *value*. It accepts and returns numeric data. In *value*, pass a positive number (greater than zero).

**Note:** **LN** is the inverse of **EXP**, so LN(EXP(x)) is x.

### Exemple

---

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

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

## LOOKUP

LOOKUP ( toFind , intervalToSearch , returnInterval )

Paramètre	Type	Description
toFind	RefCell, Texte, Numérique, Booléen	→ Value to find
intervalToSearch	RefCell	→ Cell range to search
returnInterval	RefCell	→ Cell range to find corresponding value

### Description

---

The **LOOKUP** function searches for the value *toFind* in the *intervalToSearch* cell range and returns the corresponding value used in the *returnInterval* range.

*toFind* must contain the reference of a cell that actually contains the value to find.

*intervalToSearch* must be sorted in ascending order since **LOOKUP** uses the first value higher or equal to the value set as *toFind*.

If *toFind* cannot be found, it matches the largest value in *intervalToSearch* that is less than or equal to *toFind*.

### Exemple

---

	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", located in the A8 cell, is the value to find. B1:B6 is the interval to search. C1:C6 is the return interval. The B3 cell contains the value to find. The corresponding value in the return interval is in the C3 cell, that is, "697".

## MAX

MAX ( valeur {, valeur2 , ... , valeurN} )

Paramètre	Type	Description
valeur	Réel, Tableau	→ Nombres ou tableau de nombres à évaluer

### Description

---

La fonction **MAX** retourne la plus grande (maximum) valeur de *valeur*. Elle accepte et retourne des données numériques.

Dans *valeur*, passez les valeurs à évaluer. Chaque argument peut être un nombre ou un tableau de nombre. Vous pouvez utiliser un tableau simple (une plage de cellules) ou des tableaux multiples (plusieurs plages de cellules).

Si un tableau ou une référence contient du texte, des valeurs logiques ou des cellules vides, **MAX** ignore ces valeurs ; cependant, les cellules avec la valeur zéro sont incluses dans les calculs.

### Exemple

---

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

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

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

## MID

MID ( value , startFrom {, numChars} )

Paramètre	Type	Description
value	Texte, RefCell, Expression	→ Text containing the characters to extract
startFrom	Numérique	→ Number designating the first character to extract in text
numChars	Numérique	→ Number of characters to return

### Description

---

The **MID** function returns the requested number of characters from *value* starting at the position specified in *startFrom*. It accepts text data for *value* and numeric data for *startFrom* and *numChars*. It returns text data.

In *value*, pass the text string containing the characters you want to extract. It can be a string, a formula that returns a string, or a reference to a cell containing a string.

In *startFrom*, pass a number representing the first character you want to extract in text, with the first character in the text having a value of one (1); if not an integer, the number is truncated. It can be a string, a formula that returns a string, or a reference to a cell containing a string. If *startFrom*:

- is greater than the length of *value*, an empty string (" ") is returned
- is less than the length of *value*, but *startFrom* + *numChars* exceeds the length of *value*, the characters up to the end of text are returned.

In *numChars*, pass the number of characters to return from *value*; if an integer is not specified, the number is truncated.

### Exemple

---

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

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

## MIN

MIN ( valeur {, valeur2 , ... , valeurN} )

Paramètre	Type	Description
valeur	Réel, Tableau	→ Nombre ou tableau de nombres à évaluer

### Description

---

La fonction **MIN** retourne la valeur la plus petite (minimum) de *valeur*. Elle accepte et retourne des données numériques.

Dans *valeur*, passez les valeurs à évaluer. Chaque argument peut être un nombre, ou un tableau de nombres (simple : une plage de cellule, ou multiple : des plages de cellules).

Si un tableau de référence contient du texte, des valeurs logiques, ou des cellules vides, **MIN** ignore ces valeurs ; toutefois, les cellules avec la valeur zéro sont incluses dans les calculs.

### Exemple

---

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

## NOT

NOT ( valeurLogique )

Paramètre	Type	Description
valeurLogique	Booléen, Numérique, Expression	→ Valeur à évaluer

### Description

---

La fonction **NOT** renvoie le contraire de la valeur logique passée en paramètre. *valeurLogique* peut être un booléen ou un nombre. Si sa valeur est zéro, alors la fonction retourne TRUE. Si sa valeur n'est pas zéro, alors la fonction retourne FALSE.

### Exemple

---

```
NOT(A3)
```

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

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

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

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

## **NOW**

NOW ()

Ne requiert pas de paramètre

### **Description**

---

The **NOW** function returns the time of the current date. It returns a DateTime object.  
This function is updated when the spreadsheet or cell containing the function is recalculated.

### **Exemple**

---

If today is Monday 8 January 2018 at 11:25:42:

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

## **NPER**

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

Paramètre	Type	Description
i	Numérique	→ The interest rate for a period
m	Numérique	→ The monthly payment paid at the end of the period (use a negative value). For acquired value: pass an empty parameter (see example)
p	Numérique	→ The current value of the loan
f	Numérique	→ The future value of the loan (use a negative value)

### **Description**

---

The **NPER** function returns the number of periods needed to reimburse a loan. Two formulas can be used:

- First formula, when you know the monthly payment:

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

- Second formula, when you know the total payment:

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

### **Exemple 1**

---

You borrowed 6,500 euros with 10.5% annual interest and you reimburse 166.42 euros per month:

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

### **Exemple 2**

---

You borrowed 3,000 euros with 10.5% annual interest and you know that the total monthly payments will be 5,000 euros:

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

## OR

OR ( valeurLogique {, valeurLogique2 , ... , valeurLogiqueN} )

Paramètre	Type	Description
valeurLogique	Booléen, Tableau, Expression	→ Valeur(s) à évaluer

### Description

---

La fonction **OR** retourne FALSE si l'évaluation de tous les paramètres est fausse. Elle renvoie TRUE si l'évaluation d'au moins un paramètre est vraie.

La fonction accepte les valeurs booléennes en tant qu'expressions numériques (0 ou 1) ou expression logiques (TRUE ou FALSE) jusqu'à 255 arguments. Vous pouvez également spécifier un tableau au lieu d'une liste de valeurs séparées, jusqu'à 255 tableaux. Vous pouvez aussi indiquer la valeur *valeurLogique* en tant qu'expression.

### Exemple

---

```
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
```

## **PI**

PI ()

Ne requiert pas de paramètre

### **Description**

---

La fonction **PI** renvoie la valeur de Pi, soit 3.14159265358979.

## PMT

PMT ( i , n , p )

Paramètre	Type	Description
i	Numérique	→ The interest rate for a period
n	Numérique	→ The number of periods
p	Numérique	→ The current value of the loan (use a negative value)

### Description

---

The **PMT** function returns the value of monthly loan payments.

Here is the formula for **PMT**:

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

### Exemple

---

You borrowed 6,500 euros over 48 months with 10.5% interest:

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

## ⚙️ PV

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

Paramètre	Type	Description
i	Numérique →	The interest rate for a period
n	Numérique →	The number of periods
m	Numérique →	For compound interest: the monthly payment at the end of each period (use a negative value). For single interest: pass an empty parameter (see example)
f	Numérique →	For single interest rate: the final value at the end of a period (use a negative value)

### Description

---

The **PV** function calculates:

- the current value of a sum using the compound interest, or
- the current value of a sum using single interest rates.

To calculate the current value of a sum using the compound interest, pass the *m* parameter and omit the *f* parameter. Here is the formula for this calculation:

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

To calculate the current value of a sum using single interest rates, pass the *f* parameter and pass an empty parameter (,,) for the *m* placeholder. Here is the formula for this calculation:

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

### Exemple 1

---

**Compound interest:** you have a loan with an 12% annual interest rate (thus 1% per month) over 5 months with a monthly payment of €1,000.

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

### Exemple 2

---

**Single interest rates:** you have a loan with an 12% annual interest rate (thus 1% per month) over 5 months with a monthly payment of €1,000.

**Note:** Pay attention to the double " , , " inside the syntax.

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

## **RAND**

RAND ()

Ne requiert pas de paramètre

### **Description**

---

La fonction **RAND** renvoie un nombre aléatoire compris entre 0 et 0,9999999...

### **Exemple**

---

```
RAND()
```

```
RAND()*100
```

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

## ⚙ RATE

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

Paramètre	Type	Description
n	Numérique	→ The number of periods
m	Numérique	→ The monthly payment paid at the end of the period (use a negative number). For future value: pass an empty parameter (see example)
p	Numérique	→ The current value of the loan
f	Numérique	→ The future value (use a negative number)

### Description

---

The **RATE** function returns the interest rate corresponding to the values passed in parameters. Two formulas can be used:

- First formula, when you know the monthly payment paid at the end of the period

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

- Second formula, when you know the acquired value:

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

### Exemple 1

---

You borrowed 3,000 euros and your monthly payments are 1,000 euros over 5 months:

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

### Exemple 2

---

You borrowed 2,800 euros and the total paid value is 6,000 euros over 5 months:

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

## ROUND

ROUND ( valeur , nbreDécimales )

Paramètre	Type		Description
valeur	Numérique	→	Nombre à arrondir
nbreDécimales	Numérique	→	Nombre de décimales

### Description

---

La fonction **ROUND** arrondit la *valeur* spécifiée au nombre le plus proche, en utilisant le nombre de décimales indiqué dans *nbreDécimales*.

Utilisez le paramètre *valeur* pour spécifier le nombre à arrondir. Vous pouvez passer n'importe quelle donnée numérique (ou une référence de cellule contenant une donnée numérique).

Utilisez le paramètre *nbreDécimales* pour préciser le nombre de décimales. Ce paramètre peut prendre ces valeurs :

- Fixez le nombre de décimales à une valeur >0 pour arrondir au nombre de décimales indiqué.
- Fixez le nombre de décimales à zéro pour arrondir au nombre entier le plus proche.
- Fixez le nombre de décimales à une valeur <0 pour arrondir la valeur à gauche de la décimale à la dizaine, centaine, etc. la plus proche.

### Exemple

---

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

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

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

```
ROUND(29.2,-2) //0 car 29.2 est plus proche de 0 que de 100.
```

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

## ROW

ROW ( {reference} )

Paramètre	Type		Description
reference	RefCell	→	A cell or a range of cells



### Description

---

The **ROW** function returns the row number of *reference*.

*reference* can be a cell or a range of cells. If the *reference* argument is omitted, the default argument is the reference of the cell in which the **ROW** function is placed.

### Exemple

---

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

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

## SIN

SIN ( valeur )

Paramètre	Type		Description
valeur	Réel	→	Angle dont le sinus est retourné

### Description

---

La fonction **SIN** renvoie le sinus de l'angle spécifié dans *valeur*. Elle accepte et retourne des données numériques.

Dans *valeur*, passez n'importe quel nombre réel (un angle) pour lequel on récupère le sinus. Ce nombre doit être exprimé en radians. Si l'angle est en degrés, multipliez-le par  $\text{PI}/180$  pour le convertir en radians.

### Exemple

---

```
SIN(B4)
```

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

## SQRT

SQRT ( valeur )

Paramètre	Type	Description
valeur	Numérique	→ Nombre $\geq 0$ dont on veut récupérer la racine carrée

### Description

---

La fonction **SQRT** retourne la racine carrée positive de la *valeur* spécifiée. *valeur* doit être un nombre positif (or 0), sinon une erreur est retournée.

### Exemple

---

```
SQRT(B4)
```

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

## STDEV.P

STDEV.P ( value {, value2 , ... , valueN} )

Paramètre	Type	Description
value	Numérique, RefCell	→ Values to evaluate

### Description

---

The **STDEV.P** function returns the standard deviation for an entire population of numeric values. Standard deviation is a measure of the dispersion of values in relation to the average (average value). In *value*, *value2...*, pass numeric arguments corresponding to the population. Each argument can be a cell, a cell range, or a number. This function can have up to 255 arguments.

**Note:** If your data represents a sample of the population, then compute the standard deviation using the [STDEV](#) function.

The **STDEV.P** is calculated using the "biased" or "n" method. The **STDEV.P** function uses the following formula:

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

### Exemple

---

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

STDEV.P(A1:A9)

STDEV.P(95,89,73,87,85,76,100,96,96) gives the result 8.8079649700473

## SUBSTITUTE

SUBSTITUTE ( *str* , *toReplace* , *replacement* {, *instance*} )

Paramètre	Type	Description
<i>str</i>	RefCell	→ String or reference to a cell containing the string in which you want to replace characters
<i>toReplace</i>	Texte	→ String to replace
<i>replacement</i>	Texte	→ New string to use instead of existing string
<i>instance</i>	Numérique	→ Which occurrence of the existing string to replace; otherwise every occurrence is replaced

### Description

---

The **SUBSTITUTE** function replaces *toReplace* in the *str* text with *replacement* and returns the edited text.

By default, **SUBSTITUTE** replaces the first occurrence of *toReplace*. Pass *instance* to define which occurrence you want to replace.

If no occurrence of *toReplace* is found, **SUBSTITUTE** returns *str*.

### Exemple

---

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

## SUM

SUM ( valeur {, valeur2 , ... , valeurN} )

Paramètre	Type	Description
valeur	Numérique, RefCell	→ Nombre(s) à additionner

### Description

---

La fonction **SUM** retourne la somme des cellules ou d'une sélection de cellules.

Dans *valeur*, passez les valeurs à évaluer. Chaque argument peut être un numérique ou une plage de cellules (une sélection continue ou discontinue de cellules).

**Note :** L'opérateur + fournit une auto-conversion pour les valeurs non-numériques transmises par des constantes et pour des valeurs non-numériques transmises par référence. La fonction **SUM** fournit une auto-conversion pour les valeurs non-numériques passées par constante mais ignore les valeurs non-numériques passées par référence.

### Exemple

---

```
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 ( valeur )

Paramètre	Type		Description
valeur	Réel	→	Angle dont la tangente est retournée

### Description

---

La fonction **TAN** renvoie la tangente de l'angle spécifié dans *valeur*. Elle accepte et retourne des données numériques.

Dans *valeur*, passez n'importe quel nombre réel (un angle) pour lequel on récupère la tangente. Ce nombre doit être exprimé en radians. Si l'angle est en degrés, multipliez-le par  $\text{PI}/180$  pour le convertir en radians.

### Exemple

---

```
TAN(B3)
```

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

## TEXT

TEXT ( valeur , format )

Paramètre	Type	Description
valeur	Numérique, RefCell	→ Valeur numérique à formater en texte
format	Texte	→ Format à appliquer à la valeur

### Description

---

La fonction **TEXT** retourne une chaîne composée du nombre *valeur* formaté selon le *format*.  
Passez une donnée numérique ou une référence à une cellule contenant une donnée numérique dans *valeur* et un paramètre texte dans *format*.

### Exemple

---

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

```
TEXT(100,"0.00€") // 100.00€
```

## **TODAY**

TODAY ()

Ne requiert pas de paramètre

### **Description**

---

The **TODAY** function returns the date and time of the current date. It returns a DateTime object. This function is updated when the spreadsheet or cell containing the function is recalculated.

### **Exemple**

---

If the current day is Monday 8 January 2018:

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

## TRUE

TRUE ()

Ne requiert pas de paramètre

### Description

---

La fonction **TRUE** retourne la valeur logique TRUE (1).

### Exemple

---

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

## TYPE

TYPE ( value )

**Paramètre**  
value

**Type**



**Description**  
Value to evaluate

### Description

---

The **TYPE** function returns the type of *value* as a number.

Returned types are listed below:

Type of value	Returned number
Number	1
DateTime object	1
TimeSpan object	1
Text	2
Boolean	4
Error	16
Array	64

Use the **TYPE** function when the execution of another function depends on the type of value contained in a specific cell. The **TYPE** function is particularly useful when calling functions that accept different types of data.

### Exemple

---

```
TYPE(G15)

TYPE(42) //1

TYPE("String") //2

TYPE(TRUE) //4
```

## VAR.P

VAR.P ( value {, value2 , ... , valueN} )

Paramètre	Type	Description
value	Numérique, Tableau	→ Values to get the variance

### Description

---

The **VAR.P** function returns the variance based on the entire population, which uses numeric values.

In *value*, pass the values to evaluate. Each argument can be a number, or an array of numbers (a single array (cell range) or multiple arrays (cell ranges)).

**Note:** This function assumes that its arguments are the entire population. If your data represents only a sample of the population, then compute the variance using the [VAR.S](#) function.

### Exemple

---

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

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

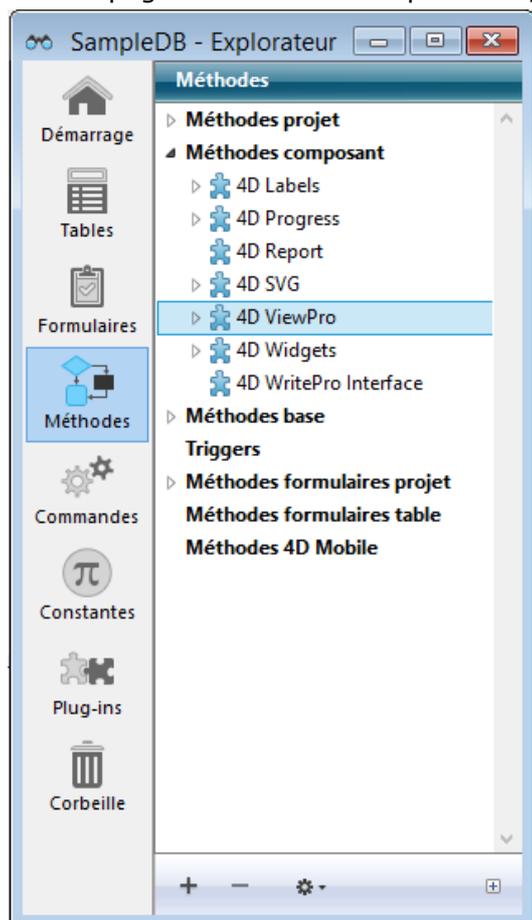
```
VAR.P(98,85,76,87,92,89,90) //39.265306122449
```

# Langage 4D View Pro

-  A propos des commandes 4D View Pro
-  Manipulation des zones 4D View Pro
-  Evénement formulaire Sur VP prêt
-  VP Convert from 4D View
-  VP EXPORT DOCUMENT
-  VP Export to object
-  VP IMPORT DOCUMENT
-  VP IMPORT FROM OBJECT
-  VP NEW DOCUMENT

## 🔧 A propos des commandes 4D View Pro

La fonctionnalité de tableur 4D View Pro est un composant interne à 4D. Ainsi, 4D View Pro apparaît dans la page Méthodes de l'Explorateur, dans la section "Méthodes composant" :



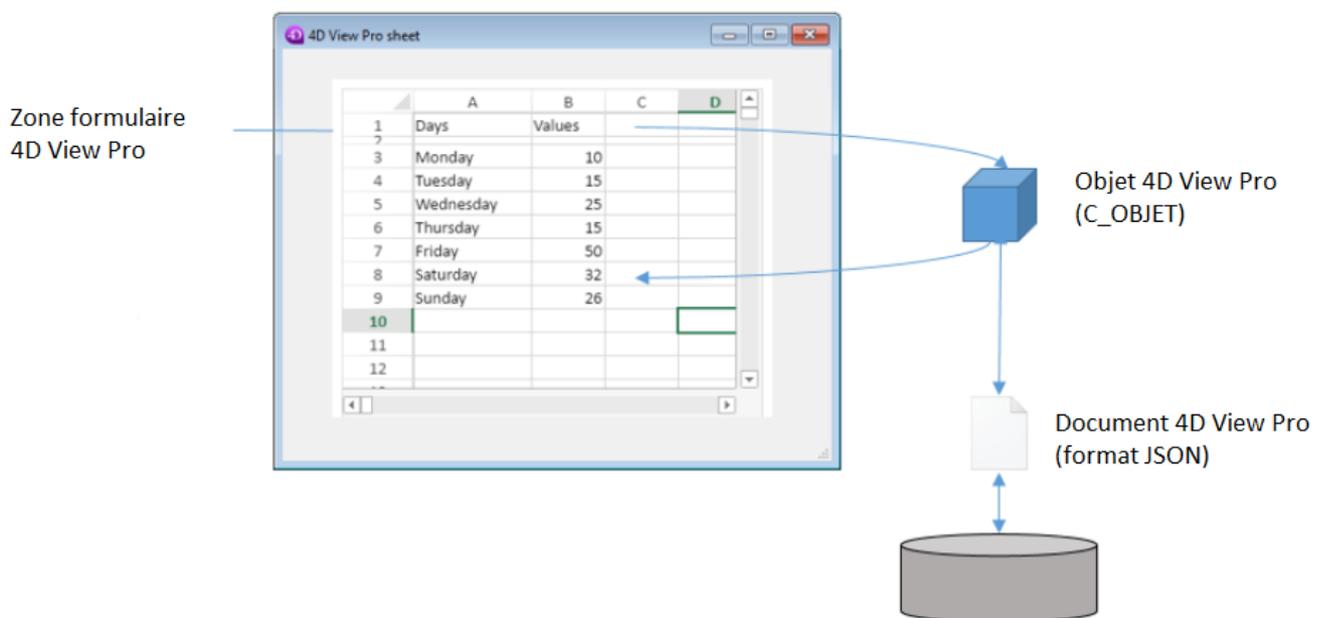
Vous pouvez déployer cet élément de façon à voir toutes les commandes du composant. Ces commandes peuvent être utilisées dans l'éditeur de méthode de 4D, comme toutes les commandes du langage 4D.

## Manipulation des zones 4D View Pro

### Architecture

Lorsque vous utilisez des zones 4D View Pro dans vos formulaires, vous devez gérer plusieurs éléments :

- La **zone formulaire 4D View Pro** (objet formulaire 4D) : contient et affiche l'objet 4D View Pro. Cette zone est définie par un nom (champ "Nom" dans la Liste des Propriétés).
- l'**objet 4D View Pro** (variable ou expression de type **C\_OBJET**) : stocke le contenu complet du tableur (voir ci-dessous). Vous associez un objet 4D View Pro à une zone formulaire 4D View Pro via le champ "Variable ou expression" de la Liste des propriétés.
- Le **document 4D View Pro** (documents .4vp) : stocke le contenu complet du tableur au format JSON.



Lorsque vous chargez un objet 4D View Pro dans une zone formulaire, 4D génère l'événement formulaire Sur\_VP\_prêt une fois que toute la zone est chargée. Vous devez exécuter tout code 4D View Pro traitant la zone dans cet événement, sinon une erreur est renvoyée.

### Objet 4D View Pro

L'objet 4D View Pro décrit le document et est géré automatiquement par les commandes 4D View Pro. Il contient les propriétés suivantes :

Propriété	Type valeur	Description
version	Entier long	Version du composant interne
dateCreation	Timestamp	Date de création
dateModified	Timestamp	Date de la dernière modification
meta	Objet	Contenu libre, réservé aux Développeurs 4D
spreadJS	Objet	Réservé pour le composant 4D View Pro

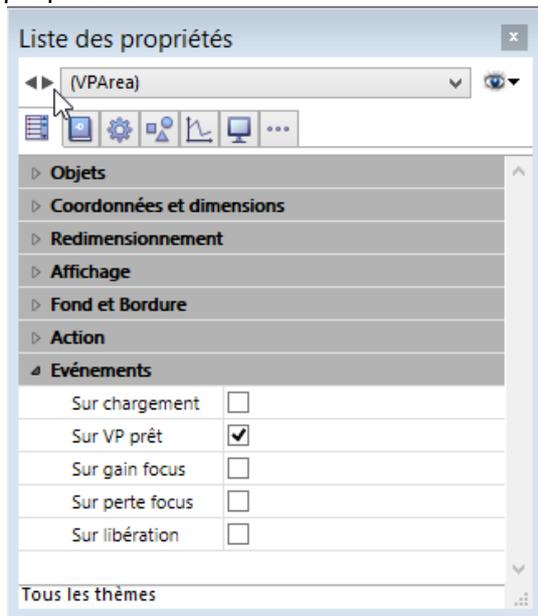
## Événement formulaire Sur VP prêt

Tout code d'initialisation d'une zone 4D View Pro, pour le chargement ou la lecture de valeurs à partir de la zone ou dans la zone, doit se situer dans l'événement formulaire Sur VP prêt de la zone. Cet **Événement formulaire** est exécuté une fois la zone entièrement chargée. En testant l'événement, vous êtes sûr que le code sera exécuté dans un contexte valide. Une erreur est retournée si une commande 4D View Pro est appelée avant que l'événement Sur VP prêt soit généré.

**Note :** les zones 4D View Pro sont chargées de façon asynchrone dans les formulaires 4D. Cela signifie que l'événement formulaire standard Sur chargement ne peut pas être utilisé pour l'initialisation du code 4D View Pro, car il pourrait être exécuté avant que le chargement de la zone ne soit terminé. Sur VP prêt est toujours généré après Sur chargement.

Pour insérer le code d'initialisation d'une zone 4D View Pro :

1. Vérifiez que l'événement **Sur VP prêt** est coché pour la zone 4D View Pro, dans la Liste des propriétés :



2. Ecrivez le code l'initialisation de la zone 4D View Pro dans l'événement Sur VP prêt :

```
// méthode formulaire ou méthode objet de la zone
Au cas ou
  :(Evenement formulaire=Sur VP prêt)
  ... // code d'initialisation
Fin de cas
```

## ⚙️ VP Convert from 4D View

VP Convert from 4D View ( Document4DView ) -> Résultat

Paramètre	Type		Description
Document4DView	BLOB	→	Document 4D View
Résultat	Objet	↩	Objet 4D View Pro

### Description

---

La commande **VP Convert from 4D View** vous permet de convertir un document 4D View existant en un objet 4D View Prot.

**Note :** Cette commande ne nécessite pas l'installation du plug-in 4D View dans votre environnement. Dans le paramètre *Document4DView*, passez une variable BLOB ou un champ contenant le document 4D View à convertir. La commande retourne un *objet* 4D View Pro dans lequel toutes les informations stockées à l'origine dans le document 4D View sont converties en attributs 4D View Pro.

La plupart des informations du document 4D View sont directement converties. Les propriétés non-prises en charge, ou les propriétés qui doivent être modifiées pendant la conversion, sont répertoriées dans la page **Conversion des documents 4D View**.

### Exemple

---

Vous souhaitez obtenir un objet 4D View Pro à partir d'une zone 4D View stockée dans un BLOB:

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

## VP EXPORT DOCUMENT

VP EXPORT DOCUMENT ( vpAreaName ; filePath )

Paramètre	Type		Description
vpAreaName	Texte	→	Nom de la zone 4D View Pro sur le formulaire
filePath	Texte	→	Chemin d'accès du document

### Description

---

La commande **VP EXPORT DOCUMENT** exporte l'objet 4D View Pro attaché à la zone 4D View Pro *vpAreaName* vers un document sur disque en fonction du paramètre *filePath*.

Passez dans *vpAreaName* le nom de la zone sur le formulaire 4D. Si vous passez un nom invalide, une erreur est retournée.

Dans *filePath*, passez le chemin de destination et le nom du document que vous souhaitez exporter. Si vous passez simplement le nom du document, il sera sauvegardé au même niveau que le fichier de structure 4D.

L'extension du document 4D View Pro est ".4vp". Elle est automatiquement ajoutée au nom du document si elle n'est pas indiquée dans *filePath*.

### Exemple

---

Vous souhaitez exporter le contenu de la zone "VPArea" vers un document 4D View Pro sur le disque :

```
C_TEXTE($docPath)
$docPath:="C:\\ViewProDocs\\MyExport.4VP"
VP EXPORT DOCUMENT("VPArea";$docPath)
//MyExport.4VP est sauvegardé sur le disque
```

## VP Export to object

VP Export to object ( vpAreaName ) -> Résultat

Paramètre	Type		Description
vpAreaName	Texte	→	Nom de la zone 4D View Pro sur le formulaire
Résultat	Objet	↩	Objet 4D View Pro

### Description

---

La commande **VP Export to object** retourne l'objet 4D View Pro attaché à la zone 4D View Pro *vpAreaName*. Vous pouvez utiliser cette commande par exemple pour stocker la zone 4D View Pro dans un champ objet de la base de données 4D.

Passez dans *vpAreaName* le nom de la zone sur le formulaire 4D. Si vous passez un nom invalide, une erreur est retournée.

Pour plus d'informations sur les objets 4D View Pro, référez-vous au paragraphe **Objet 4D View Pro**.

### Exemple

---

Vous souhaitez lire la propriété "version" de la zone 4D Write Pro courante :

```
C_OBJET($vpAreaObj)
C_ENTIER LONG($vpVersion)
$vpAreaObj:=VP Export to object("vpArea")
// $vpVersion:=OB Lire($vpAreaObj;"version")
$vpVersion:=$vpAreaObj.version
```

## VP IMPORT DOCUMENT

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

Paramètre	Type		Description
<i>vpAreaName</i>	Texte	→	Nom de la zone 4D View Pro sur le formulaire
<i>filePath</i>	Texte	→	Chemin d'accès du document

### Description

---

La commande **VP IMPORT DOCUMENT** importe le document 4D View Pro dont le chemin d'accès est indiqué dans *filePath* et l'affiche dans la zone 4D View Pro *vpAreaName*. Le document importé remplace toutes les données déjà présentes dans la zone.

Passez dans *vpAreaName* le nom de la zone sur le formulaire 4D. Si vous passez un nom invalide, une erreur est retournée.

Dans *filePath*, passez le chemin de destination et le nom du document à importer. Seuls les documents 4D View Pro (extension ".4vp") sont supportés par la commande. Vous devez passer un chemin d'accès complet, sauf si le document est localisé au même niveau que le fichier de structure de la base, dans ce cas vous passez juste son nom.

Une erreur est retournée si le paramètre *filePath* est invalide, ou si le fichier est manquant ou mal-formé.

### Exemple

---

Vous souhaitez importer un document 4D View Pro stocké sur le disque, à l'ouverture du formulaire :

```
C_TEXTE($docPath)
Si(Eventement formulaire=Sur VP prêt) //La zone 4D View Pro est chargée et prête
  $docPath:="C:\\Bases\\ViewProDocs\\MyExport.4VP"
  VP_IMPORT_DOCUMENT("VPArea";$docPath)
Fin de si
```

## VP IMPORT FROM OBJECT

VP IMPORT FROM OBJECT ( *vpAreaName* ; *viewPro* )

Paramètre	Type		Description
<i>vpAreaName</i>	Texte	→	Nom de la zone 4D View Pro sur le formulaire
<i>viewPro</i>	Objet	→	Objet 4D View Pro

### Description

---

La commande **VP IMPORT FROM OBJECT** importe l'objet 4D View Pro *viewPro* et l'affiche dans la zone *vpAreaName*. Le contenu de l'objet importé remplace toutes les données insérées auparavant dans la zone.

Passez dans *vpAreaName* le nom de la zone sur le formulaire 4D. Si vous passez un nom invalide, une erreur est retournée.

Dans *viewPro*, passez un objet 4D View Pro valide. Cet objet peut avoir été créé en utilisant **VP Export to object** ou manuellement. Pour plus d'informations sur les objets 4D View Pro, référez-vous à la section **Objet 4D View Pro**.

Une erreur est retournée si l'objet *viewPro* est invalide.

### Exemple

---

Vous souhaitez importer une feuille de calcul stockée dans un champ objet :

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

## VP NEW DOCUMENT

VP NEW DOCUMENT ( *vpAreaName* )

Paramètre	Type	Description
<i>vpAreaName</i>	Texte →	Nom de la zone 4D View Pro sur le formulaire

### Description

---

La commande **VP NEW DOCUMENT** charge et affiche un nouveau document dans l'objet *vpAreaName* de la zone 4D View Pro sur le formulaire. Le nouveau document vide remplace toutes les données auparavant insérées dans la zone.

Passez dans *vpAreaName* le nom de la zone sur le formulaire 4D. Si vous passez un nom invalide, une erreur est retournée.

### Exemple

---

Vous souhaitez afficher un document vide dans l'objet formulaire "myVPArea" :

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