

# 4D v11 SQL r2

---

*Addendum*  
*Windows®/Mac OS®*



---

## **4D v11 SQL Release 2 (11.2) - Addendum** **Versions Windows® and Mac OS®**

Copyright© 1985 - 2008 4D SAS.  
Tous droits réservés.

---

Les informations contenues dans ce manuel peuvent faire l'objet de modifications sans préavis et ne sauraient en aucune manière engager 4D SAS. La fourniture du logiciel décrit dans ce manuel est régie par un octroi de licence dont les termes sont précisés par ailleurs dans la licence électronique figurant sur le support du Logiciel et de la Documentation y afférente. Le logiciel et sa Documentation ne peuvent être utilisés, copiés ou reproduits sur quelque support que ce soit et de quelque manière que ce soit, que conformément aux termes de cette licence.

Aucune partie de ce manuel ne peut être reproduite ou recopiée de quelque manière que ce soit, électronique ou mécanique, y compris par photocopie, enregistrement, archivage ou tout autre procédé de stockage, de traitement et de récupération d'informations, pour d'autres buts que l'usage personnel de l'acheteur, et ce exclusivement aux conditions contractuelles, sans la permission explicite de 4D SAS.

4D, 4D Draw, 4D Write, 4D View, 4D Insider, 4D Developer, 4ème Dimension®, 4D Server ainsi que le logo 4D sont des marques enregistrées de 4D SAS.

Windows, Windows XP, Windows Vista et Microsoft sont des marques enregistrées de Microsoft Corporation.

Apple, Macintosh, iMac, Mac OS et QuickTime sont des marques enregistrées ou des noms commerciaux de Apple Computer, Inc.

Mac2Win Software Copyright © 1990-2008 est un produit de Altura Software, Inc.

ICU Copyright © 1995-2006 International Business Machines Corporation and others. All rights reserved.

ACROBAT © Copyright 1987-2008, Secret Commercial Adobe Systems Inc. Tous droits réservés. ACROBAT est une marque enregistrée d'Adobe Systems Inc.

4D inclut un programme développé par Apache Software Foundation (<http://www.apache.org/>).  
4D utilise des logiciels de cryptographie écrits par Eric Young (eay@cryptsoft.com), ainsi que des logiciels écrits par Tim Hudson (tjh@cryptsoft.com).

Correcteur orthographique, © Copyright SYNAPSE Développement, Toulouse, France, 1994-2008.

Tous les autres noms de produits ou appellations sont des marques déposées ou des noms commerciaux appartenant à leurs propriétaires respectifs.

# Sommaire

<b>Chapitre 1</b>	<b>Introduction</b> .....	<b>7</b>
<b>Chapitre 2</b>	<b>Mode Développement</b> .....	<b>9</b>
	Nouvel assistant de bienvenue .....	10
	Nouvelles propriétés de champs .....	11
	Incrémentation auto .....	11
	Option de stockage interne .....	12
	Nouvel algorithme de définition des mots-clés .....	13
	Nouvel attribut de méthode : Exécuter sur serveur .....	14
	Contexte d'exécution .....	15
	Pointeurs .....	16
	Exemple .....	16
	List box .....	17
	Lignes invisibles .....	17
	Saisie et affichage multilignes (Windows) .....	18
	Explorateur de ressources .....	18
	Présentation .....	19
	Utilisation .....	20
	Fonctionnement en mode client-serveur .....	26
	Zones Web .....	28
	Créer une zone Web .....	29
	Spécificités Mac OS .....	29
	Spécificités Windows .....	30
	Variables associées .....	31
	Actions standard .....	32
	Evénements formulaire .....	33
	Interface utilisateur .....	34
	Commandes du langage .....	35
	Construire une application client/serveur .....	35

Enregistrement de l'accès à une base .....	37
Création des fichiers .....	38
Utilisation .....	39
Lancement du serveur Web sous Mac OS .....	39

<b>Chapitre 3</b>	<b>Langage .....</b>	<b>41</b>
	Méthode base Sur authentification SQL .....	41
	Environnement 4D .....	43
	OUVRIR FENETRE ADMINISTRATION .....	43
	NOTIFIER MODIFICATION DOSSIER RESSOURCES .....	45
	Dossier 4D .....	45
	FIXER PARAMETRE BASE, Lire parametre base .....	47
	Environnement système .....	51
	ENREGISTRER EVENEMENT .....	51
	Images .....	52
	BLOB VERS IMAGE .....	52
	LIRE FICHER IMAGE .....	53
	List box .....	53
	LIRE TABLEAUX LISTBOX .....	53
	Outils .....	54
	OUVRIR URL WEB .....	54
	Recherches et tris .....	55
	CHERCHER PAR TABLEAU DANS SELECTION .....	55
	CHERCHER PAR FORMULE, CHERCHER PAR FORMULE DANS SELECTION .....	55
	Nouvelle préférence de compatibilité .....	56
	Sélections .....	57
	VIDER TABLE .....	57
	Trigger Sur chargement enregistrement .....	58
	Web Services (Client) .....	58
	AUTHENTIFIER WEB SERVICE .....	58
	FIXER OPTION WEB SERVICE .....	59
	Web Services (Serveur) .....	60
	DECLARATION SOAP .....	60
	Zone Web .....	62
	WA OUVRIR URL .....	62
	WA Lire URL courant .....	63
	WA OUVRIR URL PRECEDENT .....	64
	WA URL precedent disponible .....	64
	WA OUVRIR URL SUIVANT .....	65
	WA URL suivant disponible .....	65
	WA ACTUALISER URL .....	66
	WA ARRETER CHARGEMENT URL .....	66
	WA Executer JavaScript .....	66

WA EXECUTER FONCTION JAVASCRIPT	67
WA FIXER FILTRES URL	68
WA LIRE FILTRES URL	71
WA FIXER FILTRES LIENS EXTERNES	71
WA LIRE FILTRES LIENS EXTERNES	73
WA LIRE DERNIERE ERREUR URL	74
WA Lire dernier URL filtre	75
WA LIRE HISTORIQUE URL	75
WA Creer menu historique URL	76
WA Lire titre page URL	78
WA FIXER CONTENU PAGE	78
WA Lire contenu page	79
WA AGRANDIR TEXTE PAGE	79
WA REDUIRE TEXTE PAGE	80
WA FIXER PREFERENCE	80
WA LIRE PREFERENCE	81

## Chapitre 4    **4D Server v11 SQL** . . . . . **83**

Nouvelle architecture	83
Gestion des dossiers distants	84
Nouvelle fenêtre d'administration	86
Moniteur	87
Utilisateurs	90
Process	93
Maintenance	97
Serveur d'application	100
Serveur SQL	103
Serveur HTTP	104
Nouvelles fonctions accessibles à distance	107
Administration de 4D Server depuis un poste distant	107
Compilation depuis les postes distants	109
Accès aux préférences de sauvegarde depuis les postes distants	110
4D Server v11 SQL et le langage	111
Exécution des commandes par formule sur le serveur	111
Sélections temporaires et ensembles	113
Configuration du serveur	113
Numéros de ports TCP	113
Accès au Centre de sécurité et de maintenance	114

<b>Chapitre 5</b>	<b>Plug-ins</b> .....	<b>115</b>
	4D Internet Commands .....	115
	SMTP_SetPrefs .....	115
	4D View .....	116
	Figer et libérer les volets .....	116
	Nouvelles commandes .....	118
	PV FIGER VOLETS .....	118
	PV LIBERER VOLETS .....	119
	Modifications du langage .....	120

# 1

# Introduction

Bienvenue dans la release 2 de 4D v11 SQL.

Cette nouvelle version marque la disponibilité de **4D Server v11 SQL**, permettant aux applications 4D client/serveur de tirer parti de la puissance et les performances du moteur de 4D v11 SQL. En outre, 4D Server v11 SQL bénéficie d'une nouvelle fenêtre d'administration ergonomique et d'importantes optimisations.

De nombreuses nouvelles fonctions ont également été ajoutées dans 4D v11 SQL r2, s'appliquant aussi bien aux applications monopostes que client/serveur : nouvelles propriétés de champs et de méthodes, nouvel Explorateur de ressources, zones Web incluses dans les formulaires... Le langage intégré de 4D propose aussi de multiples nouveautés et modifications.

Enfin, les versions 11.2 des plug-ins 4D Internet Commands et 4D View incluent également de nouvelles fonctions.

Toutes ces nouveautés sont détaillées dans les chapitres suivants :

- Mode Développement
- Langage
- 4D Server v11 SQL
- Plug-ins



# 2

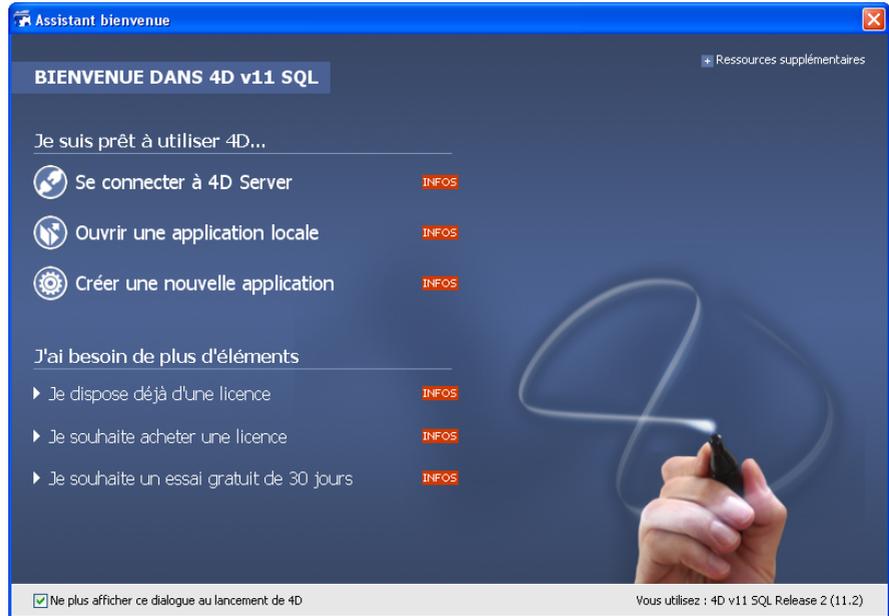
## Mode Développement

Cette section présente les nouveautés et modifications apportées à l'environnement de développement de 4D v11 SQL, tant au niveau de la base de données qu'à celui des outils RAD et des formulaires :

- nouvel assistant de bienvenue
- nouvelles propriétés de champs,
- nouvel attribut de méthode "Exécuter sur serveur",
- lignes invisibles et saisie multiligne dans les list box,
- Explorateur de ressources,
- nouveaux objets "Zones Web",
- nouveautés liées à la construction d'applications client/serveur,
- nouveau mode d'enregistrement des accès aux bases,
- nouveau mode de lancement du serveur Web sous Mac OS.

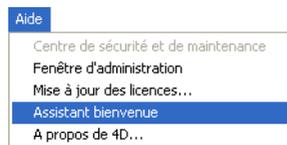
## Nouvel assistant de bienvenue

4D bénéficie d'un nouvel assistant de bienvenue, apparaissant par défaut au premier lancement de l'application 4D :



Cet assistant est principalement destiné à guider les nouveaux utilisateurs de 4D au démarrage du programme. Il donne accès à toutes les actions possibles, qu'elles soient relatives à l'utilisation ou à l'activation de l'application. Des informations supplémentaires sont fournies lorsque l'utilisateur clique sur le bouton **INFOS** associé à une action.

Par défaut, l'assistant est affiché à chaque démarrage. Vous pouvez le masquer en cochant l'option **Ne plus afficher ce dialogue au lancement de 4D**. Vous pourrez à tout moment l'afficher de nouveau (et désélectionner l'option) en sélectionnant la commande **Assistant bienvenue** dans le menu **Aide** :



# Nouvelles propriétés de champs

La version 11.2 de 4D v11 SQL propose deux nouvelles propriétés applicables aux champs de la base de données.

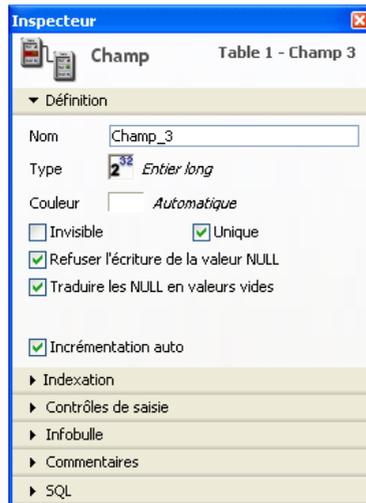
**Incrémentation auto** 4D propose désormais la propriété de champ "Incrémentation auto", permettant de générer des identifiants uniques pour les enregistrements.

Un champ auquel cette propriété a été affectée est incrémenté automatiquement à chaque création d'enregistrement dans la table. Il est géré uniquement par le moteur de 4D. Il n'est ni saisissable ni modifiable par l'utilisateur ou par programmation. Si vous tentez de le modifier, une erreur est générée.

Les numéros sont attribués définitivement et ne sont jamais réutilisés, même en cas de suppression d'enregistrements. Les numéros générés durant une transaction annulée sont "perdus".

Cette propriété est utilisable avec les champs de type Entier, Entier long et Entier long 64. Elle correspond à l'attribut SQL "AUTO\_INCREMENT" et peut être définie via ce langage (cf. exemple).

Côté 4D, l'attribut "Incrémentation auto" est visible et modifiable dans l'Inspecteur de l'éditeur de Structure :



Le libellé AUTOINCREMENT apparaît également dans la zone d'information "SQL" de l'Inspecteur pour les champs disposant de cet attribut.

Il est possible de définir plusieurs champs "Incrémentation auto" dans une même table. Un compteur est associé individuellement à chaque champ. Lorsque la propriété est supprimée pour un champ, le compteur est réinitialisé à 0.

- ▶ Le code suivant permet de générer un champ "id" de type entier dans la table "animals" :

**TABLEAU TEXTE(\$names;6)**

\$names{1}:="dog"

\$names{2}:="cat"

\$names{3}:="penguin"

\$names{4}:="snake"

\$names{5}:="butterfly"

\$names{6}:="ostrich"

**Debut SQL**

```
CREATE TABLE animals(  
  id INT NOT NULL AUTO_INCREMENT ,  
  name VARCHAR(30) NOT NULL,  
  PRIMARY KEY(id));  
INSERT INTO animals(name) VALUES(:$names);
```

**Fin SQL**

Le code suivant :

```
SELECT * FROM animals;
```

... retourne ces valeurs :

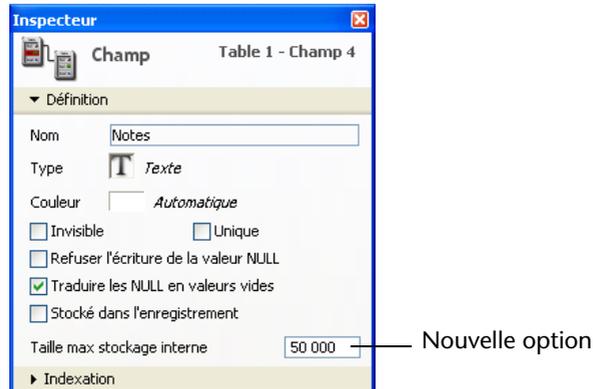
id	name
1	dog
2	cat
3	penguin
4	snake
5	butterfly
6	ostrich

### Option de stockage interne

Dans les versions précédentes de 4D v11 SQL, pour des raisons d'optimisation, les données des champs de type BLOB, Image et Texte (option) étaient toujours stockées en-dehors des enregistrements.

La version 11.2 vous permet désormais de choisir de stocker ou non ces données en-dehors de vos enregistrements en fonction de leur taille. Si votre application traite des données BLOB, Image ou texte de taille limitée, il peut être intéressant de les inclure dans les enregistrements.

Ce paramétrage est accessible via une nouvelle option de l'Inspecteur de champs :



Saisissez une valeur personnalisée dans la zone "Taille max stockage interne". Cette valeur représente une taille en octets au-dessous de laquelle les données du champ seront stockées dans l'enregistrement. Par exemple, si vous saisissez 30 000 pour un champ image, une image de 20 ko sera stockée dans l'enregistrement et une image de 40 ko sera stockée en-dehors de l'enregistrement.

Par défaut, la valeur est 0 : toutes les données sont stockées en-dehors des enregistrements.

## Nouvel algorithme de définition des mots-clés

A compter de la version 11.2, 4D utilise un nouvel algorithme basé sur la bibliothèque ICU pour déterminer la liste des mots dans un champ texte lors de la construction d'un index de mots-clés. Le découpage en mots est désormais plus précis et tient compte des spécificités linguistiques.

Les différences suivantes par rapport au précédent algorithme sont à noter :

- les nombres sont considérés dans leur ensemble, incluant les éventuels symboles décimaux,
- les mots contenant des apostrophes et des tirets sont considérés dans leur ensemble.

---

*Note* Pour plus d'informations, reportez-vous à la page <http://www.icu-project.org/userguide/boundaryAnalysis.html>.

---

L'exemple suivant illustre les différences de définition de mots-clés entre les versions précédentes de 4D et la version 11.2. Le texte à indexer est "le prix d'aujourd'hui est 11.42€" :

Ancien algorithme (4D v11.0, 11.1)	Nouvel algorithme (4D v11.2 et suivantes)
le	le
prix	prix
d	aujourd'hui
aujourd	est
hui	11.42
est	
11	
42	

---

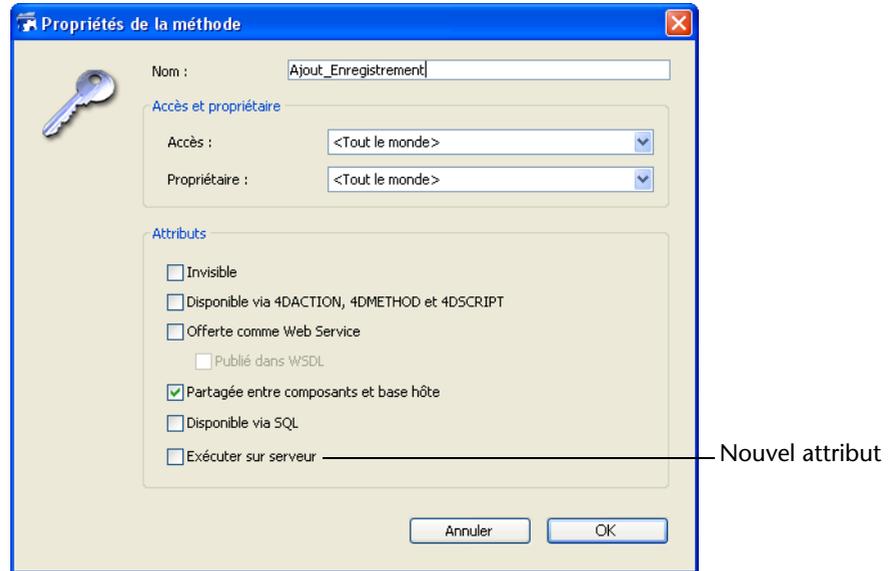
**IMPORTANT :** *Si vous avez déjà créé des index de mots-clés à l'aide d'une version 11.0 ou 11.1 de 4D, vous devez impérativement réindexer les champs concernés avec la nouvelle version de 4D pour que l'index fonctionne correctement.*

---

## Nouvel attribut de méthode : Exécuter sur serveur

4D v11 SQL Release 2 propose un nouvel attribut pour les méthodes projet : **Exécuter sur serveur**. Cet attribut peut être défini dans la boîte

de dialogue de modification globale des attributs ou dans la boîte de dialogue des propriétés de la méthode :



Cet attribut est pris en compte uniquement dans le cadre d'une application 4D en client/serveur.

Lorsque cette option est cochée, la méthode projet est toujours exécutée sur le serveur, quel que soit le mode d'appel de la méthode.

## Contexte d'exécution

Le contexte d'exécution de la méthode projet est le même que celui des triggers : la méthode sur le serveur partage le même contexte de base de données pour le verrouillage d'enregistrements et les transactions que le contexte correspondant côté client.

Tous les paramètres de la méthode (\$1, \$2, etc.) sont envoyés sur le serveur et la valeur du paramètre \$0, s'il est utilisé, est retournée sur le client.

A la différence de la commande Exécuter sur serveur, cette option ne provoque pas de création de process sur le serveur. 4D Server utilise le process "jumeau" du process client qui a demandé l'exécution. En outre, l'option simplifie le principe de délégation de l'exécution d'une méthode sur le serveur car le transfert des paramètres s'effectue automatiquement dans les deux sens, comme pour un appel de méthode "normal".

La commande Executer sur serveur, elle, a un fonctionnement asynchrone et requiert donc davantage de programmation et le recours aux sémaphores pour la lecture des résultats.

### Pointeurs

Si vous passez un pointeur sur une variable (variable simple, tableau ou élément de tableau), la valeur pointée est également envoyée sur le serveur. Si la valeur pointée est modifiée sur le serveur par la méthode, la valeur modifiée est retournée sur le client pour mise à jour de la variable correspondante côté client.

Les pointeurs sur une table ou un champ sont envoyés sous forme de référence (numéro de table, numéro de champ). La valeur de l'enregistrement courant n'est pas échangée automatiquement.

---

*Note* L'option fonctionne de la même manière en mode interprété comme en mode compilé.

---

### Exemple

Voici le code la méthode projet *Monappli* ayant l'attribut "Exécuter sur serveur" :

```
C_POINTEUR($1) `Pointeur sur table
C_POINTEUR($2) `Pointeur sur champ
C_POINTEUR($3) `Pointeur sur tableau
C_TEXTE($4) `Valeur à rechercher
C_ENTIER LONG($0) `Résultat

`Rechercher et rapatrier des valeurs pour chaque enregistrement
CHERCHER($1-> ; $2-> = $4)
Tant que(Non(Fin de selection($1->)))
  AJOUTER A TABLEAU($3-> ; maFormule($1))
  ENREGISTREMENT SUIVANT($1->)
Fin tant que
LIBERER ENREGISTREMENT($1->)
$0 := Enregistrements trouves( $1->)
```

Côté client, l'appel de la méthode s'effectue ainsi :

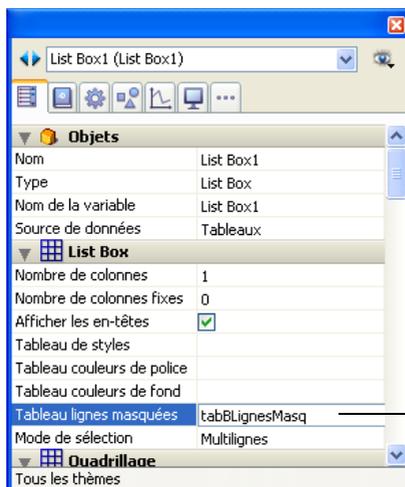
```
TABLEAU TEXTE(monTab; 0)
$vlnombre := MonAppli ( ->[Table_1] ; ->[Table_1]Champ_1 ; ->monTab ;
"à trouver")
```

# List box

## Lignes invisibles

Il est désormais possible de masquer une ou plusieurs ligne(s) dans les list box de type tableau.

Le principe de définition des lignes masquées est semblable à celui des styles ou des couleurs : la nouvelle propriété **Tableau lignes masquées** du thème "List box" permet d'associer un tableau à la list box.



Définition des lignes masquées

Vous devez saisir le nom d'un tableau de type Booléen. Le tableau doit avoir le même nombre d'éléments que la list box. Chaque élément de ce tableau représente le statut affiché/masqué de la ligne correspondante dans la list box :

- Pour masquer la ligne, passez Vrai,
- Pour afficher la ligne, passez Faux.

S'il n'y a pas de tableau défini ou si le tableau est vide, toutes les lignes de la list box sont visibles. Par défaut, les lignes ajoutées sont visibles.

Masquer des lignes affecte uniquement l'affichage de la list box. Les lignes masquées sont toujours présentes dans les tableaux et peuvent être manipulées par programmation. Les commandes du langage, notamment Lire nombre lignes listbox ou LIRE POSITION CELLULE LISTBOX, ne tiennent pas compte de l'état masqué/affiché des lignes. Par exemple, dans une list box contenant 10 lignes et dont les 9 premières sont masquées, Lire nombre lignes listbox retournera 10.

Du point de vue de l'utilisateur, la présence de lignes masquées dans une list box n'est pas décelable visuellement. Seules les lignes visibles sont sélectionnables (par exemple via la commande **Tout sélectionner**).

La commande LIRE TABLEAUX LISTBOX permet de lire le tableau de visibilité (cf. [commande LIRE TABLEAUX LISTBOX](#), page 53).

### Saisie et affichage multilignes (Windows)

Dans la version Windows de 4D v11 SQL r2, vous pouvez saisir et afficher du texte sur plusieurs lignes au sein d'une même "cellule" de list box.

Pour ajouter un retour à la ligne, il suffit d'appuyer sur les touches **Ctrl+Retour chariot** :

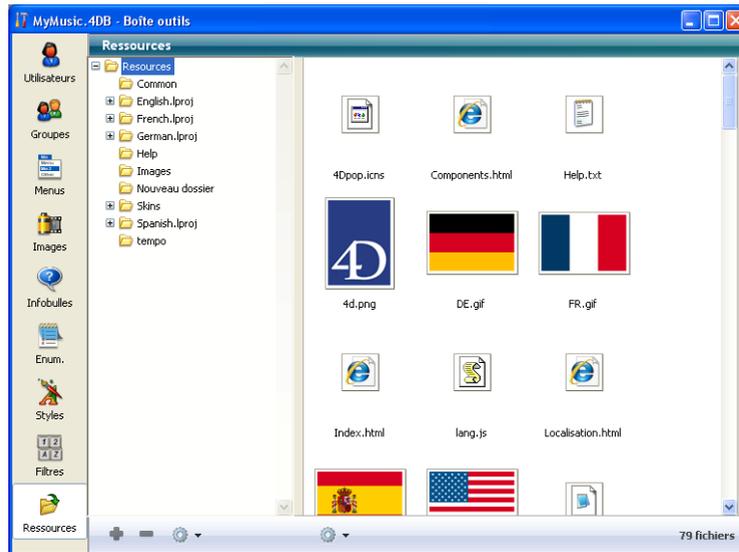
Nom	Prénom	Ville
Electra	Omar	Nevers
Elizabeth	Oliver	Paris, TX (USA)

A noter que la hauteur des lignes n'est pas redimensionnée automatiquement.

## Explorateur de ressources

Dans 4D v11 SQL, l'architecture des ressources est basée sur le dossier **Resources**, situé à côté du fichier de structure de la base (.4db ou .4dc). Ce dossier permet de stocker toutes les "ressources" d'une base ou d'un composant, c'est-à-dire tous les fichiers nécessaires à la traduction, à la personnalisation de l'interface de l'application (fichiers image, texte, XLIFF...) ou à son fonctionnement en général.

Afin notamment de faciliter la gestion de ce dossier en architecture client-serveur, la version 11.2 de 4D v11 SQL comporte un nouvel outil permettant de gérer le contenu du dossier Ressources : l'Explorateur de ressources. Cet outil est accessible dans la Boîte à outils de 4D :



## Présentation

L'explorateur de ressources affiche sous forme de liste hiérarchique le contenu du dossier Ressources de la base courante. Il comporte en outre plusieurs fonctions supplémentaires facilitant la gestion du contenu de ce dossier : ajout et suppression d'éléments, prévisualisation, etc.

L'explorateur de ressources peut être utilisé avec 4D en mode local ou en mode distant.

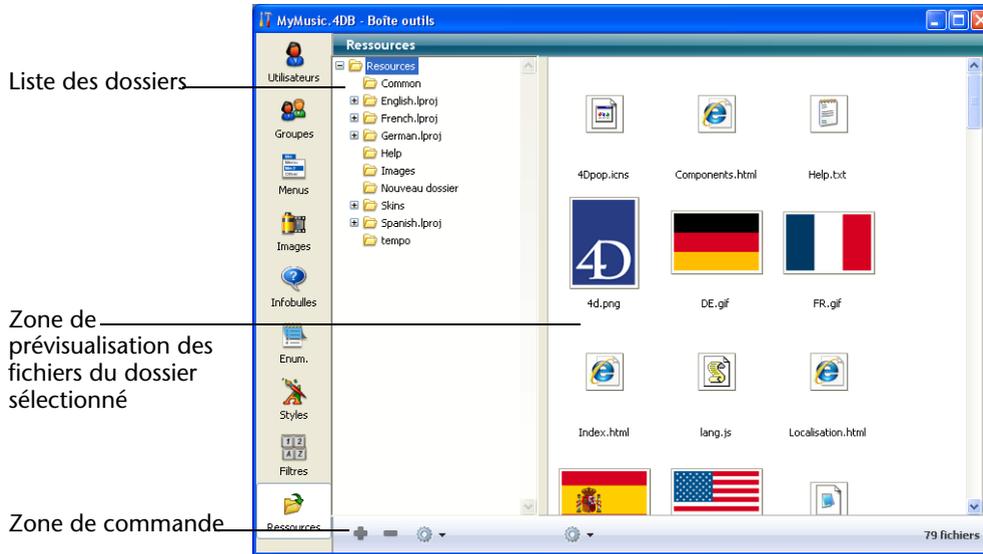
C'est en mode distant que l'explorateur de ressources présente le plus d'intérêt : en effet, dans ce contexte il permet de contrôler la synchronisation du contenu du dossier Ressources entre tous les postes distants connectés à 4D Server. Autrement dit, l'explorateur de ressources permet de gérer le "partage" des ressources en environnement client-serveur.

Un mécanisme de notification permet d'informer les postes clients de la modification du contenu du dossier **Ressources**. Chaque poste client peut alors se synchroniser avec le serveur.

## Utilisation

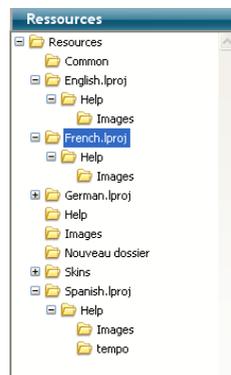
L'explorateur de ressources permet de visualiser et de modifier le contenu du dossier Resources de la base courante. Il est doté de diverses commandes pour l'ajout, la suppression, la recherche et la visualisation des éléments de ressources.

L'explorateur de ressources contient deux zones distinctes : la liste des dossiers et la zone de visualisation, chacune étant munie de boutons de commande :



## Liste des dossiers

La liste des dossiers affiche la hiérarchie des dossiers présente dans le dossier Resources de la base.

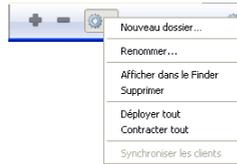


Vous pouvez sélectionner, déployer et contracter des dossiers. Lorsque vous cliquez sur un nom de dossier, les fichiers qu'il contient éventuellement sont affichés dans la zone de prévisualisation (partie

droite de l'explorateur de ressources).

Au bas de la fenêtre se trouve une zone de commande, comportant des boutons et des menus d'action.

La zone de commande située sous la liste des dossiers contient les boutons [+] et [-] ainsi qu'un menu d'action.



Le bouton [+] crée un dossier dans le dossier sélectionné ou au niveau principal si aucun dossier n'est sélectionné. Le bouton [-] supprime le dossier sélectionné ainsi que son contenu.

---

*Note* Vous pouvez également utiliser un menu contextuel avec la liste des dossiers. Les commandes de ce menu sont identiques à celles du menu d'action.

---

Les commandes du menu d'action sont les suivantes :

- **Nouveau dossier...** : crée un dossier dans le dossier sélectionné ou au niveau principal si aucun dossier n'est sélectionné. Lorsque vous choisissez cette commande, une boîte de dialogue s'affiche, vous permettant de saisir le nom du dossier à créer. Comme le dossier sera créé physiquement sur le disque, veillez à ce que le nom donné ne contienne pas de caractères non autorisés par le système (tels que : ou /).
- **Renommer...** : affiche une boîte de dialogue permettant de renommer le dossier sélectionné.

---

*Note* Il n'est pas possible de renommer le dossier Ressources lui-même.

---

- **Montrer sur disque** (actif uniquement en mode local) : affiche le dossier dans une fenêtre du système d'exploitation.
- **Supprimer** : supprime le dossier sélectionné et son contenu.

---

*Note* Il n'est pas possible de supprimer le dossier Ressources lui-même.

---

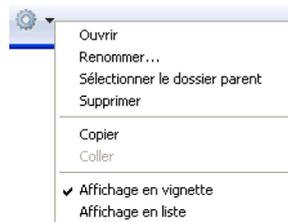
- **Déployer tout/Contracter tout** : déploie ou contracte tous les dossiers de la liste.

- **Aviser les clients** (actif uniquement en mode distant) : permet de "forcer" l'envoi aux autres postes clients d'une notification de mise à jour du contenu du dossier Ressources. Les autres clients pourront alors effectuer une synchronisation immédiate ou différée de leur dossier Ressources en fonction de leurs propres paramètres (cf. [paragraphe "Mise à jour des postes clients", page 27](#)).  
Utilisez cette commande lorsque vous avez effectué une modification du dossier Ressources et souhaitez demander immédiatement la synchronisation des autres clients.
- **Mise à jour des ressources** (actif uniquement en mode distant) : permet de "forcer" la synchronisation du dossier Ressources local avec celui du poste serveur.  
Utilisez cette commande lorsque vous avez été notifié d'une mise à jour du dossier Ressources du poste serveur (cf. [paragraphe "Notification des postes clients", page 26](#)). Vous pouvez également forcer la mise à jour globale du contenu du dossier Ressources local (téléchargement du dossier du serveur et remplacement du dossier local) en maintenant la touche **Majuscule** enfoncée lors de la sélection de la commande **Mise à jour des ressources**.

### Zone de visualisation

La partie droite de la fenêtre de l'explorateur de ressources est une zone de visualisation affichant les fichiers contenus dans le dossier sélectionné (y compris les fichiers présents dans des éventuels sous-dossiers). Cette zone est mise à jour à chaque modification de l'élément sélectionné dans la liste des dossiers.

Cette zone bénéficie également de commandes spécifiques, accessibles via le menu d'action situé sous la zone :



- **Ouvrir** : ouvre le fichier sélectionné dans l'éditeur par défaut, s'il a été défini. Cette action correspond à un double-clic sur le fichier.
- **Renommer** : affiche une boîte de dialogue permettant de renommer le fichier sélectionné.

- **Sélectionner le dossier parent** : sélectionne dans la Liste des dossiers le dossier parent du fichier sélectionné. Cette commande permet de visualiser précisément l'emplacement du fichier dans la hiérarchie du dossier Ressources. En effet, la zone de visualisation affiche la totalité des fichiers présents dans le dossier sélectionné, y compris les fichiers présents dans les éventuels sous-dossiers.
- **Supprimer** : supprime le fichier sélectionné.
- **Copier** : copie le fichier sélectionné dans le Presse-papiers.
- **Coller** : colle le contenu du Presse-papiers dans le dossier sélectionné si le Presse-papiers contient une image ou un chemin de fichier. Si le Presse-papiers contient une image, 4D créera un fichier image du type correspondant. Une boîte de dialogue s'affiche, vous permettant de nommer le fichier créé.
- **Affichage en vignette / Affichage en liste** : ces commandes fonctionnent en bascule.
  - En mode **Affichage en vignette**, les fichiers du dossier sélectionné sont affichés sous forme de vignettes. Les fichiers de type image sont prévisualisés (si le format de l'image est reconnu par 4D), les autres types de fichiers apparaissent avec leur icône système.

Affichage en vignette



Dans ce mode de visualisation, un menu contextuel donne accès à des commandes d'édition des fichiers (cf. ci-dessous).

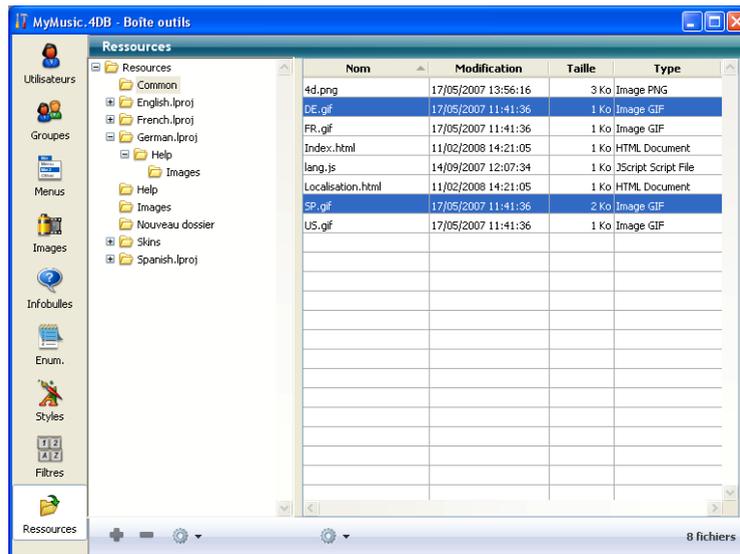
Une info-bulle associée à chaque image fournit des informations

supplémentaires : chemin d'accès (relatif), nom, type, date et heure de modification et la taille.



- En mode **Affichage en liste**, les fichiers du dossier sélectionné sont affichés sous forme de tableau. Pour chaque fichier, le tableau fournit directement le nom, la date et l'heure de modification, la taille et le type :

Affichage en liste



Vous pouvez trier le tableau en cliquant dans la zone d'en-tête d'une colonne.

- **Menu contextuel**

Lorsque les fichiers sont affichés en vignette, la zone bénéficie d'un menu contextuel :

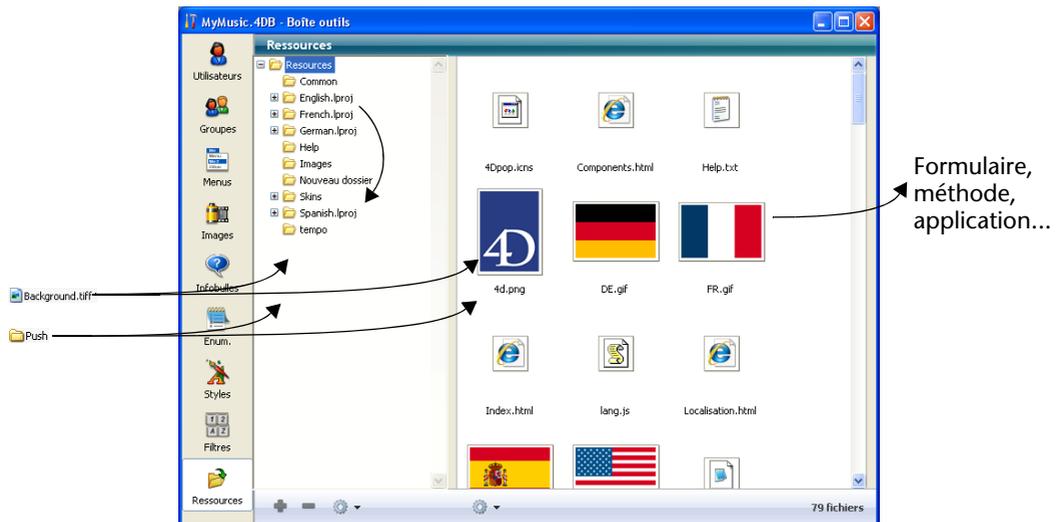


Les commandes de ce menu sont identiques à celles du menu d'action, à la différence que seules les commandes pertinentes en fonction du contexte sont affichées.

## Gestion par glisser/déposer

L'explorateur de ressources vous permet d'ajouter des éléments dans le dossier Ressources et ses sous-dossiers à l'aide du glisser-déposer :

- Vous pouvez modifier l'arborescence du dossier Ressources par glisser-déposer des dossiers à l'intérieur de la liste des dossiers.
- Vous pouvez également ajouter des fichiers dans le dossier Ressources par glisser-déposer depuis le bureau du système d'exploitation. 4D crée automatiquement dans le dossier Ressources une copie des fichiers déposés. Vous pouvez déposer les fichiers dans la liste des dossiers ou directement dans la zone de prévisualisation.
- Il est possible de déposer des dossiers dans le dossier Ressources. Le dossier peut provenir du système d'exploitation ou du dossier Ressources d'une autre application 4D. Si le dossier glissé provient d'une autre application 4D, il est déplacé, sinon 4D en crée une copie, (comme pour les fichiers).  
Vous pouvez déposer un dossier dans la liste des dossiers ou directement dans la zone de prévisualisation. Dans ce dernier cas, le dossier est créé à l'emplacement correspondant au niveau hiérarchique affiché dans la zone de prévisualisation.
- Vous pouvez glisser-déposer des images provenant du dossier Ressources dans l'éditeur de formulaires et de méthodes. L'image est alors insérée sous forme de référence.



**Fonctionnement en mode client-serveur** L'explorateur de ressources facilite le développement collaboratif en environnement client/serveur. Il permet notamment de gérer la synchronisation en temps réel du contenu du dossier Ressources sur tous les postes connectés.

---

*Note* Comme dans les versions précédentes de 4D, pour les clients non connectés, la synchronisation du dossier Ressources est effectuée automatiquement à la connexion.

---

Les principales étapes de la synchronisation sont les suivantes :

- 1- mise à jour du dossier Ressources sur le serveur en cas de modification sur un poste client,
- 2 - information de tous les clients connectés,
- 3 - mise à jour en temps réel des poste clients connectés.

**Mise à jour du serveur** Le principe de fonctionnement dans ce mode est le suivant : toute modification effectuée en local dans le dossier Ressources via l'explorateur de ressources est automatiquement reportée sur le serveur.

**Notification des postes clients** Les postes clients connectés sont informés que le contenu du dossier Ressources a été modifié :

- soit automatiquement par le serveur, deux minutes après la dernière modification effectuée par un client (cette temporisation permet d'éviter les notifications intempestives en cas de copie de nombreux fichiers).
- soit manuellement via la commande **Aviser les clients** dans le menu d'action de l'explorateur de ressources (cf. [paragraphe "Liste des dossiers", page 20](#)) sur le poste client à l'origine de la modification.
- soit par programmation, via la nouvelle commande **NOTIFIER MODIFICATION DOSSIER RESSOURCES**. Cette commande est utile en cas de modification du contenu du dossier Ressources sur le poste serveur via une procédure stockée.

Côté client, le mode de prise en compte de l'information de modification dépend des paramètres effectués sur chaque poste (cf. paragraphe suivant).

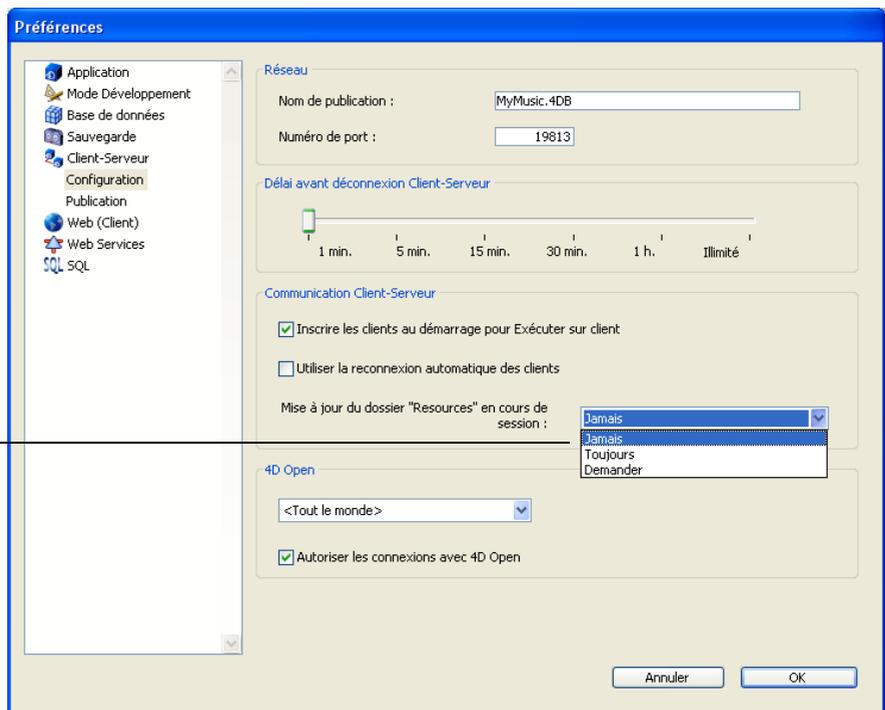
## Mise à jour des postes clients

Une fois qu'il a été "avisé" de la modification du contenu du dossier Ressources sur le serveur, chaque poste client peut se synchroniser. Cette synchronisation peut être soit automatique soit manuelle, en fonction des préférences définies globalement pour la base ou individuellement pour un poste client.

Ce paramétrage peut être défini globalement via les Préférences ou individuellement par poste client et par session à l'aide de la commande `FIXER PARAMETRE BASE`.

La définition du mode de mise à jour s'effectue via l'option "Mise à jour du dossier 'Ressources' en cours de session" dans la page **Client-Server/Configuration** des Préférences :

Configuration de la mise à jour dynamique du dossier Ressources



Trois paramétrages sont proposés :

- **Jamais** : Le dossier Ressources local n'est pas mis à jour en cours de session. La notification envoyée par le serveur est ignorée. Le dossier Ressources local pourra être mis à jour manuellement via la commande **Mise à jour des ressources** (cf. [paragraphe "Liste des dossiers"](#), page 20).

- **Toujours** : la synchronisation du dossier Ressources local est automatiquement effectuée en cours de session lorsque la notification est envoyée par le serveur.
- **Demander** : Lorsque la notification est envoyée par le serveur, une boîte de dialogue s'affiche sur les postes clients, signalant la modification. L'utilisateur peut accepter ou refuser la synchronisation du dossier Ressources local.

---

*Note* Si le paramétrage est effectué dans les Préférences sur le serveur, il sera appliqué à tous les postes clients. S'il est effectué sur un poste client, il s'appliquera à ce poste uniquement.

---

Pour plus d'informations sur le paramétrage du mode de synchronisation via la commande `FIXER PARAMETRE BASE`, reportez-vous au [paragraphe "FIXER PARAMETRE BASE, Lire parametre base"](#), page 47.

## Zones Web

La version 11.2 de 4D vous permet d'ajouter un nouveau type de zones dans vos formulaires : les **zones Web** (*Web Areas*). Ces zones peuvent afficher tout type de contenu Web à l'intérieur même de votre environnement 4D : pages HTML au contenu statique ou dynamique, fichiers, images, Javascript, Flash, PDF<sup>1</sup>... et même des documents MS Office (sous Windows lorsque la suite MS Office est installée).

---

*Note* Cette fonctionnalité remplace *4D Live Window* présent dans le pack 4D Web 2.0.

---

Plusieurs nouvelles actions standard et de nombreuses commandes de langage permettent au développeur de contrôler le fonctionnement des zones Web. Des variables spécifiques permettent d'échanger des informations entre la zone et l'environnement 4D. Vous pouvez ainsi bénéficier d'un navigateur Web basique dans vos formulaires.

A l'exécution, les zones Web peuvent également bénéficier d'un menu contextuel et de fonctions de glisser-déposer standard.

---

1. Pour afficher des documents PDF, Mac OS utilise l'outil Preview (inclus dans le système) alors que Windows nécessite l'installation d'Acrobat Reader.

Le moteur de rendu de la zone Web dépend de la plate-forme d'exécution de l'application :

- sous Mac OS X, 4D utilise le moteur WebKit d'Apple (rendu du navigateur Safari),
- sous Windows, 4D utilise le contrôle ActiveX Web (rendu du navigateur Internet Explorer).

**Créer une zone Web** La création d'une zone Web s'effectue à l'aide d'une nouvelle variante du bouton Zone de plug-in/Sous-formulaire de la barre d'objets de l'éditeur de formulaires de 4D :



Pour ajouter une zone Web dans votre formulaire, sélectionnez le bouton **Zone Web** et dessinez la zone. Il est possible de créer plusieurs zones Web dans un même formulaire.

Comme les autres objets dynamiques de formulaire, une zone Web dispose d'un nom d'objet et d'un nom de variable, vous permettant de l'adresser par programmation. La variable standard associée à l'objet zone Web est de type Texte. Vous pouvez en particulier utiliser les commandes CHOIX VISIBLE et DEPLACER OBJET avec les zones Web.

*Note* La variable texte associée à la zone Web ne contient pas de référence et ne peut donc pas être passée en tant que paramètre à une méthode. Par exemple, pour une zone Web nommée *MaZone*, le code suivant ne peut pas être utilisé :

```
Mamethode(MaZone)
```

Code de *Mamethode* :

```
WA ACTUALISER URL($1) `Ne fonctionne pas
```

Pour ce type de programmation, vous devez utiliser des pointeurs :

```
Mamethode(->MaZone)
```

Code de *Mamethode* :

```
WA ACTUALISER URL($1->) `Fonctionne
```

**Spécificités Mac OS** L'utilisation de zones Web sous Mac OS requiert des conditions spécifiques liées au système : le protocole doit obligatoirement être inclus dans l'URL demandé et la fenêtre contenant la zone doit avoir le mode "compositing".

### Protocole inclus dans l'URL

Les URLs manipulés par programmation dans les zones Web sous Mac OS doivent débiter par le protocole. Par exemple, vous devez passer la chaîne "http://www.monsite.fr" et non uniquement "www.monsite.fr".

### Mode compositing

Pour pouvoir être affichées, les zones Web doivent être incluses dans des fenêtres dessinées en "mode compositing". Ce mode interne de gestion des fenêtres sous Mac OS n'est pas utilisé dans toutes les fenêtres de 4D.

Dans 4D v11 SQL, les fenêtres dessinées en "mode compositing" sont :

- toutes les fenêtres générées par la commande Créer fenetre formulaire ;
- les fenêtres générées par la commande Créer fenetre ayant le *type* Mode compositing (constante de valeur 4096) ;
- en mode Développement (exécution via le bouton ) , les fenêtres affichant un formulaire projet.

---

*Note* Certains objets d'ancienne architecture ne sont pas compatibles avec le mode compositing (par exemple les zones 4D Chart). S'ils sont affichés dans des fenêtres en mode compositing, ces objets ne fonctionneront pas.

---

## Spécificités Windows

### Conflit Zone Web et serveur Web

Sous Windows, il est déconseillé d'accéder via une zone Web au serveur Web de l'application 4D contenant la zone car cette configuration peut provoquer un conflit paralysant l'application. Bien entendu, un 4D distant peut accéder au serveur Web du 4D Server, mais pas à son propre serveur Web.

### Documents MS Office

Sous Windows, les zones Web peuvent prendre en charge l'affichage et la modification de documents Microsoft Office (lorsque Microsoft Office est installé sur le poste). En particulier, les documents Word, Excel et Powerpoint (extensions .doc, .xls et .ppt) peuvent être traités. Le format XML MS Office est également pris en charge.

---

*Note* MS Office 2007 ne permet pas par défaut l'affichage de documents dans un navigateur Web, ils sont toujours ouverts dans une nouvelle fenêtre. Vous pouvez modifier ce fonctionnement à l'aide des instructions fournies à cette adresse :  
<http://support.microsoft.com/kb/162059/en-us>

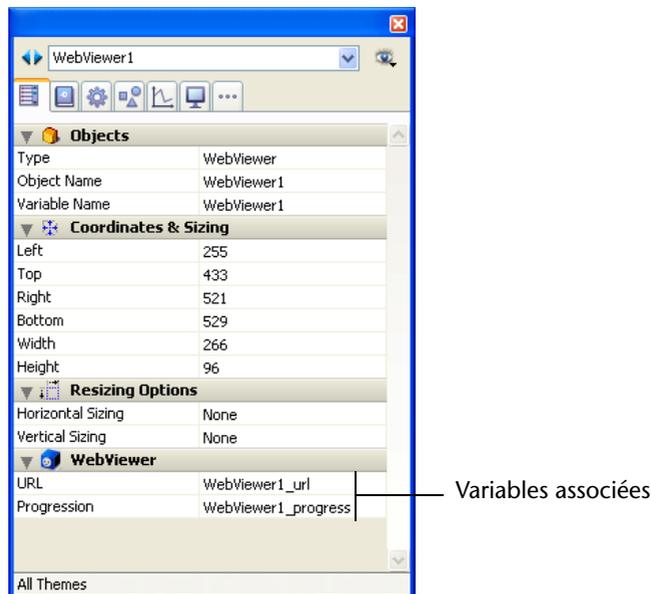
---

Enfin sous Windows, les zones Web permettent d'afficher des dossiers locaux ou externes via le protocole *ftp://* ou via les chemins réseau (*\\monserveur\monvolume*).

**Variables associées** Outre la variable objet standard (cf. [paragraphe "Créer une zone Web", page 29](#)), deux variables spécifiques sont automatiquement associées à chaque zone Web :

- la variable "URL"
- la variable "Progression du chargement"

Par défaut, ces variables sont nommées respectivement *nomZone\_url* et *nomZone\_progress*. Vous pouvez modifier ces noms comme vous le souhaitez. Les variables sont accessibles dans la Liste des propriétés :



### Variable URL

La variable "URL" est de type chaîne. Elle contient l'URL chargé ou en cours de chargement par la zone Web associée.

L'association entre la variable et la zone Web s'effectue dans les deux sens :

- Si l'utilisateur affecte un nouvel URL à la variable, l'URL est automatiquement chargé par la zone Web.
- Toute navigation effectuée à l'intérieur de la zone Web met automatiquement à jour le contenu de la variable.

Schématiquement, cette variable fonctionne comme la zone d'adresse d'un navigateur Web. Vous pouvez la représenter par une zone de texte située au-dessus de la zone Web.

### ■ Variable URL et commande WA OUVRIR URL

La variable URL produit les mêmes effets que la commande WA OUVRIR URL (cf. [commande WA OUVRIR URL, page 62](#)). Les différences suivantes sont toutefois à noter :

- pour les accès aux documents, la variable accepte uniquement des URLs conformes aux RFC ("file://c:/Mon%20Doc") et non les chemins d'accès système ("c:\MonDoc"). La commande WA OUVRIR URL accepte les deux notations.
- si la variable URL contient une chaîne vide, la zone Web ne tente pas de charger l'URL. La commande WA OUVRIR URL génère une erreur dans ce cas.
- si la variable URL ne contient pas de protocole (http, mailto, file, etc.), la zone Web ajoute "http://", ce qui n'est pas le cas pour la commande WA OUVRIR URL.
- lorsque la zone Web n'est pas affichée dans le formulaire (lorsqu'elle se trouve sur une autre page du formulaire), l'exécution de la commande WA OUVRIR URL est sans effet tandis que la valorisation de la variable URL permet de mettre à jour l'URL courant.

### Variable Progression du chargement

La variable "Progression du chargement" est de type Entier long. Elle contient une valeur entre 0 et 100, représentant le pourcentage du chargement complet de la page affichée dans la zone Web.

La variable est mise à jour automatiquement par 4D. Il n'est pas possible de la modifier manuellement.

### Actions standard

Plusieurs nouvelles actions standard sont disponibles pour la gestion automatique des zones Web. Ces actions peuvent être associées à des boutons ou des commandes de menus et permettent d'implémenter rapidement des interfaces Web basiques.

- **Ouvrir URL précédent** : cette action provoque l'ouverture de l'URL précédent parmi la séquence de navigation effectuée par l'utilisateur dans la zone Web. S'il n'y a pas d'URL précédent, c'est-à-dire si l'utilisateur n'a affiché qu'une seule page dans la zone, le bouton ou la commande de menu associé(e) est désactivé(e).

- **Ouvrir URL suivant** : cette action provoque l'ouverture de l'URL suivant parmi la séquence de navigation effectuée par l'utilisateur dans la zone Web. S'il n'y a pas d'URL suivant, c'est-à-dire si l'utilisateur n'a jamais effectué de retour en arrière dans la séquence, le bouton ou la commande de menu associé(e) est désactivé(e).
- **Actualiser URL courant** : cette action recharge le contenu courant de la zone Web.
- **Arrêter chargement URL** : cette action stoppe le chargement de la page et/ou des objets présents à l'URL courant dans la zone Web.

## Événements formulaire

De nouveaux événements formulaires ont été ajoutés dans 4D afin de permettre aux développeurs de contrôler différents aspects des zones Web. De plus, les zones Web sont compatibles avec plusieurs événements formulaires existants.

### Nouveaux événements

De nouveaux événements formulaires sont disponibles pour les zones Web.

- **Sur début chargement URL**

Cet événement est généré au début du chargement d'un nouvel URL dans la zone Web. La variable "URL" associée à la zone Web vous permet de connaître l'URL en cours de chargement.

---

*Note* L'URL en cours de chargement est différent de l'URL courant (reportez-vous à la description de la commande [WA Lire URL courant](#)).

---

- **Sur chargement ressource URL**

Cet événement est généré à chaque chargement d'une nouvelle ressource (image, frame, etc.) dans la page Web courante.

La variable "Progression du chargement" associée à la zone vous permet de connaître l'état courant du chargement.

- **Sur fin chargement URL**

Cet événement est généré lorsque toutes les ressources de l'URL courant ont été chargées.

Vous pouvez appeler la commande [WA Lire URL courant](#) afin de connaître l'URL chargé.

- **Sur erreur chargement URL**

Cet événement est généré lorsqu'une erreur a été détectée au cours du chargement d'un URL.

Vous pouvez appeler la commande [WA LIRE DERNIERE ERREUR URL](#) afin d'obtenir des informations sur l'erreur.

### ■ Sur filtrage URL

Cet événement est généré lorsque le chargement d'un URL a été bloqué par la zone Web du fait d'un filtre mis en place via la commande **WA FIXER FILTRES URL**.

Vous pouvez alors connaître l'URL bloqué à l'aide de la commande **WA Lire dernier URL filtre**.

### ■ Sur ouverture lien externe

Cet événement est généré lorsque le chargement d'un URL a été bloqué par la zone Web et que l'URL a été ouvert avec le navigateur courant du système, du fait d'un filtre mis en place via la commande **WA FIXER FILTRES LIENS EXTERNES**.

Vous pouvez alors connaître l'URL bloqué à l'aide de la commande **WA Lire dernier URL filtre**.

### ■ Sur refus ouverture fenêtre

Cet événement est généré lorsque l'ouverture d'une fenêtre pop up a été bloquée par la zone Web. En effet, les zones Web 4D ne permettent pas l'ouverture de fenêtres pop up.

Vous pouvez alors connaître l'URL bloqué à l'aide de la commande **WA Lire dernier URL filtre**.

## Événements compatibles

Les événements formulaires suivants peuvent être utilisés avec les zones Web :

- Sur chargement
- Sur libération
- Sur gain focus
- Sur perte focus
- Sur glisser
- Sur déposer
- Sur début glisser

## Interface utilisateur

Lors de l'exécution du formulaire, l'utilisateur dispose des fonctions d'interface standard des navigateurs dans la zone Web, ce qui lui permet d'interagir avec les autres zones du formulaire :

- **commandes du menu Edition** : lorsque la zone Web a le focus, les commandes du menu **Edition** permettent d'effectuer les actions de copier, coller, tout sélectionner, etc., en fonction de la sélection.

- **menu contextuel** : il est possible d'associer un menu contextuel standard à la zone Web via la Liste des propriétés. L'affichage de ce menu peut également être contrôlé via la commande [WA FIXER PREFERENCE](#)).
- **glisser-déposer** : l'utilisateur peut effectuer des glisser-déposer de textes, d'images ou de documents à l'intérieur d'une zone Web ou entre une zone Web et les objets des formulaires 4D, en fonction des propriétés des objets 4D.

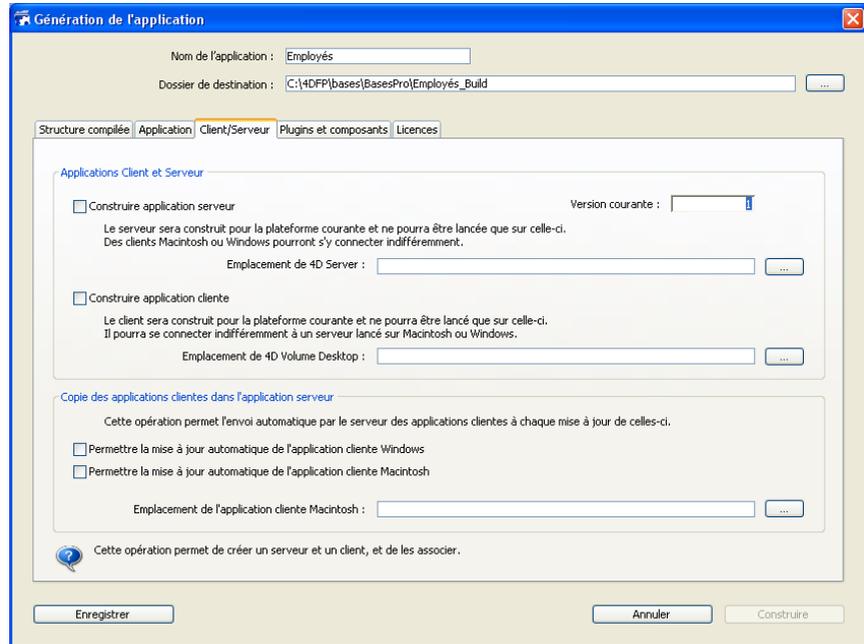
## Commandes du langage

Plus de vingt nouvelles commandes permettent de gérer les zones Web par programmation. Ces commandes sont placées dans le nouveau thème "Zone Web". Elles sont décrites dans le [paragraphe "Zone Web"](#), [page 62](#).

## Construire une application client/serveur

Comme dans les versions précédentes, 4D v11 SQL vous permet de construire des applications client/serveur personnalisées, multi-plateformes et avec option de mise à jour automatique. Cette opération doit être effectuée dans 4D en mode local.

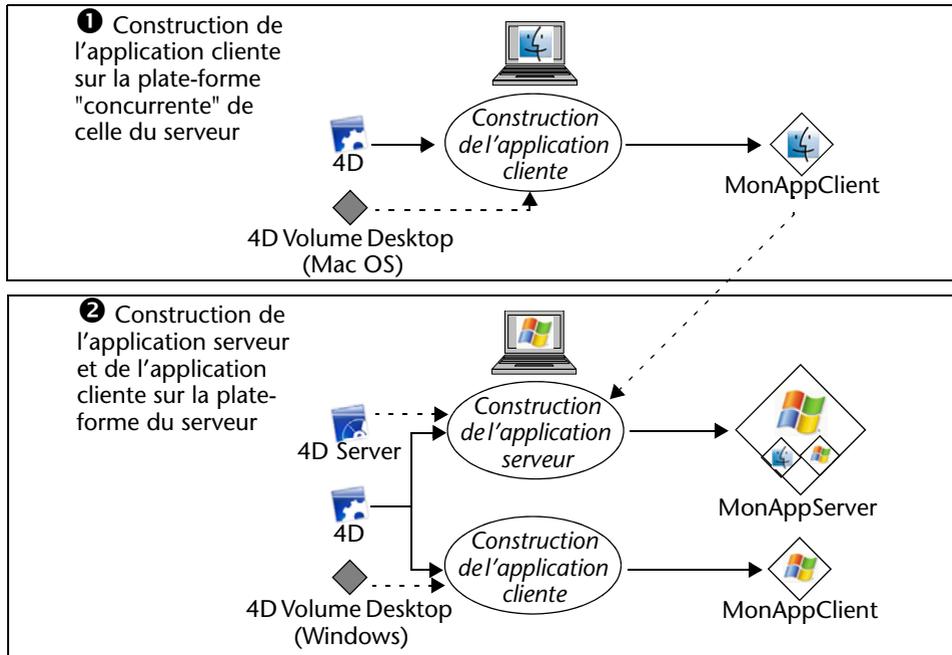
L'interface du générateur d'applications a été modifiée afin de refléter l'évolution de l'architecture des applications 4D :



- **Construire application serveur** : cochez cette option pour générer la partie serveur de votre application lors de la phase de construction. Vous devez désigner l'emplacement de l'application 4D Server sur votre disque. Vous pouvez également définir une **Version courante** afin de contrôler les connexions distantes en fonction de leur numéro de version.
- **Construire application cliente** : cochez cette option pour générer la partie cliente de votre application lors de la phase de construction. A la différence des versions précédentes de 4D, vous devez désigner l'emplacement de votre **4D Volume Desktop** pour pouvoir effectuer cette construction. En effet, 4D Volume Desktop v11 contient les éléments nécessaires à la création d'une application cliente.
- **Permettre la mise à jour automatique de l'application cliente Windows / Macintosh** : cochez ces options pour que votre application client/serveur bénéficie du mécanisme de mise à jour automatique des parties clientes via le réseau.

Si vous souhaitez créer une application cliente multi-plate-forme, vous devez désigner sur votre disque l'emplacement du dossier (progiciel sous Mac OS) de l'application cliente correspondant à la plate-forme "concurrente". Par exemple, si vous construisez votre application sous Windows, vous devez désigner à l'aide du bouton [...] le dossier du progiciel de l'application client générée sous Mac OS. Bien entendu, ce principe suppose que vous avez déjà effectué une génération de l'application cliente sur la plate-forme "concurrente".

Génération d'une application client-serveur multi-plate-forme avec mise à jour auto (serveur Windows)



## Enregistrement de l'accès à une base

4D v11 SQL permet de générer des **fichiers d'accès aux bases** contenant des paramètres destinés à automatiser et simplifier l'ouverture ou la connexion à des bases 4D. Typiquement, un fichier d'accès peut enregistrer l'adresse d'une base distante ainsi que les identifiants de connexion, évitant ainsi à l'utilisateur plusieurs opérations.

Les fichiers d'accès permettent soit l'ouverture d'une base locale soit la connexion à une base publiée par 4D Server.

*Note de compatibilité* Dans les versions précédentes de 4D, cette fonction était limitée à l'enregistrement des accès aux bases 4D Server. Elle était accessible via un bouton dans la boîte de dialogue d'identification de 4D Client. Les fichiers générés comportaient l'extension .pth. Ces fichiers ne sont plus pris en charge par 4D v11 SQL.

---

**Création des fichiers** Les fichiers d'accès de 4D v11 SQL sont des fichiers XML. Ils comportent l'extension ".4DLink". 4D génère et utilise ce type de fichier pour construire le sous-menu des "bases récentes" : un fichier .4DLink est automatiquement généré par 4D à la première ouverture d'une base locale ou à la connexion à un serveur.

Les fichiers .4DLink automatiquement créés par 4D sont placés dans le dossier des préférences locales de l'utilisateur. Dans ce dossier, deux répertoires sont créés : **Local** et **Remote**. Le dossier **Local** contient les fichiers ".4DLink" permettant de se connecter aux bases locales, le dossier **Remote** contient les fichiers ".4DLink" permettant de se connecter aux bases distantes.

Les dossiers de préférences locales sont situés à l'emplacement suivant :

- *Windows Vista* :  
C:\Users\NomUtilisateur\AppData\Roaming\4D\Favorites v11\
  - *Windows XP* : C:\Documents and Settings\NomUtilisateur\Application Data\4D\Favorites v11\
    - *Mac OS* : Users/NomUtilisateur/Library/Preferences/4D/Favorites v11/

Les fichiers présents dans ces répertoires sont affichés par 4D dans le sous-menu **Ouvrir bases récentes** du menu **Fichier** :



Les fichiers ".4DLink" peuvent aussi être créés avec tout éditeur XML et contenir diverses informations personnalisées telles que les identifiants de connexion ou le mode d'ouverture de la base.

4D fournit une DTD décrivant les clés XML utilisables pour construire un fichier ".4DLink". Cette DTD est nommée *database\_link.dtd* et est située dans le sous-dossier **\Ressources\DTD\** de l'application 4D.

## Utilisation

Un fichier d'accès .4DLink permet de lancer l'application 4D et d'ouvrir la base 4D cible. Il peut être utilisé de trois façons différentes :

- via un double-clic ou un glisser-déposer sur l'application 4D,
- via la sélection dans la boîte de dialogue d'ouverture de 4D,
- via le sous-menu **Ouvrir bases récentes** (fichier situé dans le dossier de préférences local).

Un même fichier .4DLink de type "base distante" (Remote) peut être copié et utilisé sur différentes machines.

## Lancement du serveur Web sous Mac OS

Sous Mac OS X, l'utilisation des ports TCP/IP réservés à la publication Web (ports 0 à 1023) requiert des privilèges d'accès spécifiques : seul l'utilisateur "root" de la machine peut lancer une application utilisant ces ports.

Dans les versions précédentes de 4D, le principe mis en oeuvre par défaut consistait à ouvrir une session "root" temporaire au lancement du serveur Web. Toutefois, suite à des modifications de la politique de sécurité apportée aux dernières versions de Mac OS X (Leopard), ce mécanisme ne peut plus être utilisé.

A compter de la version 11.2 de 4D, le principe consiste à déléguer l'ouverture du port à une application spécialisée, nommée *HelperTool*, disposant des privilèges suffisants.

Cette application est incluse dans le progiciel 4D. Elle doit être installée à un emplacement spécifique du système. L'installation s'effectue automatiquement lors de la première ouverture d'un port <1024 sur le poste. L'utilisateur est informé qu'un outil va être installé et est invité à saisir un nom et un mot de passe d'administrateur de la machine. Cette opération n'a lieu qu'une seule fois.

L'application est renommée "com.4D.HelperTool" et est installée dans le dossier "/Library/PrivilegedHelperTools/". Après la séquence initiale, le serveur Web de 4D peut être démarré et stoppé de façon transparente.

---

*Note* Ce mécanisme nécessite au minimum la version 10.4.6 de Mac OS X. Si vous disposez d'une version antérieure du système, vous devez utiliser une autre solution de publication.

---

# 3 Langage

Cette section regroupe les nouveautés et modifications apportées au langage de programmation de 4D v11 SQL.

## Méthode base Sur authentification SQL

La nouvelle méthode base Sur authentification SQL permet de filtrer les requêtes adressées au serveur SQL intégré de 4D. Le filtrage peut être effectué sur la base du nom, du mot de passe ainsi que (facultativement) de l'adresse IP de l'utilisateur. Le développeur peut utiliser sa propre table d'utilisateurs ou celle des utilisateurs 4D pour évaluer les identifiants de connexion. Une fois la connexion validée, l'emploi de la commande CHANGER UTILISATEUR COURANT permet de contrôler les accès de la requête au sein de la base 4D.

Lorsqu'elle existe, la méthode base Sur authentification SQL est automatiquement appelée par 4D ou 4D Server à chaque connexion externe au serveur SQL. Le système interne de gestion des utilisateurs de 4D n'est alors pas activé. La connexion est acceptée si la méthode base retourne Vrai dans \$0 et est rejetée sinon.

---

*Note* L'instruction ODBC LOGIN(SQL\_INTERNAL;\$utilisateur;\$motdepasse) ne déclenche pas l'appel de la méthode base Sur authentification SQL car il s'agit dans ce cas d'une connexion interne.

---

La méthode base reçoit jusqu'à trois paramètres de type Texte, passés par 4D (\$1, \$2 et \$3), et doit retourner un booléen, \$0. Voici la description de ces paramètres :

Paramètres	Type		Description
\$1	Texte	←	Nom d'utilisateur
\$2	Texte	←	Mot de passe
\$3	Texte	←	(Facultatif) Adresse IP du client à l'origine de la requête
\$0	Booléen	→	Vrai = requête acceptée Faux = requête rejetée

Vous devez déclarer ces paramètres de la manière suivante :

` Méthode base Sur authentification Web

**C\_TEXTE(\$1;\$2;\$3)**

**C\_BOOLEEN(\$0)**

` Code pour la méthode

Le mot de passe (\$2) est reçu en texte standard.

Vous devez contrôler les identifiants de la connexion SQL dans la méthode base Sur authentification SQL. Par exemple, vous pouvez contrôler le nom et le mot de passe à l'aide d'une table d'utilisateurs personnalisée. Si les identifiants sont valides, passez Vrai dans \$0 pour accepter la connexion et donc la requête. 4D ouvre alors une session SQL pour l'utilisateur.

Sinon, passez Faux dans \$0, dans ce cas la connexion est rejetée.

Par défaut, \$0 vaut Faux. Si la méthode base Sur authentification SQL existe et si \$0 n'est pas défini, toutes les connexions sont donc rejetées.

---

*Note* Si la méthode base Sur authentification SQL n'existe pas, la connexion est évaluée à l'aide du système intégré de gestion des utilisateurs de 4D (s'il est actif, c'est-à-dire si un mot de passe a été attribué au Super\_Utilisateur).

---

Une fois connexion acceptée, vous pouvez appeler la commande CHANGER UTILISATEUR COURANT dans la méthode base Sur authentification SQL. L'utilisation de cette commande est conseillée car elle permet un niveau de sécurité plus élevé. Ce principe d'authentification virtuelle a le double avantage de permettre le contrôle des actions de la connexion et de masquer pour l'extérieur les identifiants de la connexion dans la session SQL 4D.

Lorsque le système intégré de mots de passe 4D n'est pas activé, l'exécution de la commande CHANGER UTILISATEUR COURANT est sans effet, les utilisateurs sont connectés avec les droits du Super\_Utilisateur.

- Cet exemple de méthode base Sur authentification SQL vérifie que la demande de connexion provient du réseau interne, valide les identifiants puis affecte les droits d'utilisateur "sql\_user" pour la session SQL.

```

C_TEXTE($1;$2;$3)
C_BOOLEEN ($0)
  ` $1 : utilisateur
  ` $2 : mot de passe
  `{ $3 : Adresse IP du client}
APPELER SUR ERREUR ("SQL_error")
Si (checkInternallP($3))
  `La méthode checkInternallP vérifie que l'adresse IP est interne
  Si ($1="victor") & ($2="hugo")
    CHANGER UTILISATEUR COURANT("sql_user";"")
    Si (OK=1)
      $0:=Vrai
    Sinon
      $0:=Faux
    Fin de si
  Sinon
    $0:=Faux
  Fin de si
Sinon
  $0:=Faux
Fin de si

```

## Environnement 4D

### OUVRIR FENETRE ADMINISTRATION

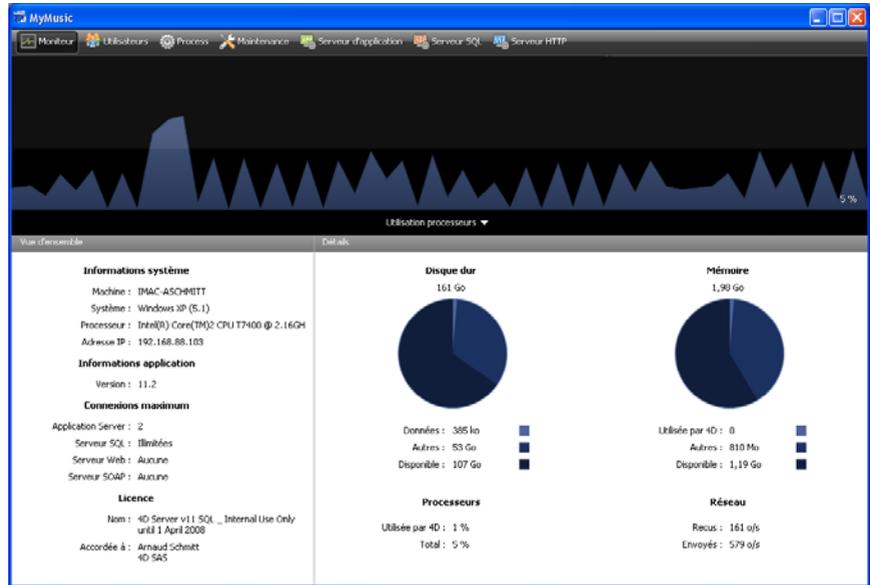
OUVRIR FENETRE ADMINISTRATION

Paramètres	Type	Description
------------	------	-------------

Cette commande ne requiert pas de paramètre

La nouvelle commande [OUVRIR FENETRE ADMINISTRATION](#) affiche la fenêtre d'administration du serveur sur le client 4D distant qui l'exécute. La fenêtre d'administration de 4D Server permet de visualiser les paramètres courants et d'effectuer diverses opérations de maintenance (cf. [paragraphe "Nouvelle fenêtre d'administration"](#),

page 86). A compter de la version 11 de 4D Server, cette fenêtre peut être affichée depuis un poste client.



Cette commande doit être appelée dans le contexte d'une application 4D connectée à un 4D Server. Elle ne fait rien si :

- elle est appelée dans une application 4D en mode local ou exécutée en procédure stockée sur le serveur,
- elle est exécutée par utilisateur autre que le Super\_Utilisateur ou l'Administrateur (dans ce cas, l'erreur -9991 est générée),

Si la commande a été exécutée correctement, la variable système OK prend la valeur 1. Dans le cas contraire, elle prend la valeur 0.

► Cet exemple pourrait être associé à un bouton d'administration :

```
Si (Type application=4D mode local)
  OUVRIRE CENTRE DE SECURITE
  ,
  ...
```

Fin de si

```
Si (Type application=4D mode distant)
  OUVRIRE FENETRE ADMINISTRATION
  ,
  ...
```

Fin de si

```
Si (Type application=4D Server)
  ,
  ...
```

Fin de si

## NOTIFIER MODIFICATION DOSSIER RESSOURCES

### NOTIFIER MODIFICATION DOSSIER RESSOURCES

Paramètres	Type	Description
		Cette commande ne requiert pas de paramètre

La nouvelle commande NOTIFIER MODIFICATION DOSSIER RESSOURCES permet de "forcer" l'envoi par 4D Server d'une notification indiquant à tous les postes 4D connectés que le contenu du dossier *Ressources* de la base a été modifié. Pour plus d'informations sur la gestion du dossier *Ressources*, reportez-vous au [paragraphe "Explorateur de ressources"](#), page 18.

Seule l'information de modification est envoyée, chaque poste distant réagira en fonction des préférences définies en local : pas de synchronisation, synchronisation automatique ou affichage d'une alerte (cf. [paragraphe "Mise à jour des postes clients"](#), page 27).

Cette commande permet en particulier de gérer la synchronisation des dossiers *Ressources* des postes distants à l'issue de la modification de ce dossier via une procédure stockée sur le serveur.

## Dossier 4D

Dossier 4D ({dossier}{; }{\*}) → Chaîne

La commande Dossier 4D accepte deux nouveaux sélecteurs et de nouveaux principes régissent l'emplacement du *Dossier 4D actif* et du *Dossier Licenses*.

### Nouveau sélecteur

La commande Dossier 4D accepte un nouveau sélecteur dans le paramètre *dossier*. La constante suivante a été ajoutée dans le thème "Environnement 4D" :

Constante	Type	Valeur
Dossier Logs	Entier long	7
Dossier racine HTML	Entier long	8

- **Dossier Logs** : Lorsque vous passez cette constante, Dossier 4D retourne le chemin d'accès du dossier qui centralise les fichiers d'historique de la base courante. Ce dossier, nommé *Logs*, se trouve au même niveau que le fichier de structure.

Le dossier Logs contient les fichiers d'historique suivants :

- conversion de la base,
- requêtes du serveur Web,

- vérification et réparation des données,
  - vérification et réparation de la structure,
  - journal d'activités sauvegarde/restitution,
  - débogage des commandes,
  - requêtes 4D Server (généralisé sur les clients et sur le serveur).
- **Dossier racine HTML** : Lorsque vous passez cette constante, Dossier 4D retourne le chemin d'accès du dossier racine HTML courant de la base. Le dossier racine HTML est le dossier dans lequel le serveur Web de 4D va chercher les pages et fichiers Web demandés. Par défaut, il est nommé *DossierWeb* et est placé à côté de fichier de structure (4D en mode local et 4D Server) ou de l'application 4D (4D en mode distant). Son emplacement peut être modifié dans la page Web/Configuration des préférences ou dynamiquement via la commande `FIXER RACINE HTML`.
- Si la commande Dossier 4D est appelée depuis un 4D distant, le chemin retourné est celui du poste distant, pas celui de 4D Server.

### Emplacement du dossier 4D actif

La commande **Dossier 4D** avec le sélecteur *Dossier 4D actif* retourne le dossier des préférences utilisateurs.

Dans les versions précédentes, ce dossier pouvait également stocker les licences et son emplacement pouvait varier suivant le type d'application 4D ou les installations successives de 4D.

Désormais, le dossier des préférences utilisateur contient les préférences de 4D v11, les dossiers Macros v2 et Favorites v11 ainsi que le fichier `shortcuts.xml` et est toujours situé à l'emplacement suivant :

- (Windows Vista) `C:\Users\{nom_utilisateur}\AppData\Roaming\4D`
- (Windows XP) `c:\Documents and settings\{nom_utilisateur}\Application Data\4D`
- (Mac OS X) `/users/{nom_utilisateur}/library/Preferences/4D`

### Emplacement du dossier Licenses

L'emplacement du *Dossier Licenses* est désormais le suivant :

- (Windows Vista) `C:\ProgramData\4D\Licenses`
- (Windows XP) `c:\Documents and settings\all users\Application Data\4D`
- (Mac OS X) `/library/application support/4D/licences`

Dans le cas d'une application fusionnée avec un 4D Volume Desktop, le dossier des licences est inclus dans le *package* de l'applicatif.

Comme dans les versions précédentes de 4D, si le dossier des licences n'a pu être créé dans le système à cause d'un défaut d'autorisation, il est créé dans :

- (Windows Vista) C:\Users\{nom\_utilisateur}\AppData\Roaming\4D
- (Windows XP) c:\Documents and settings\{nom\_utilisateur}\Application Data\4D
- (Mac OS X) /users/{nom\_utilisateur}/library/application support/4D

*Note* Les dossiers de cache pour les connexions distantes (contenant les éléments téléchargés depuis la base serveur) sont maintenant toujours créés dans

- (Windows Vista) C:\Users\{nom\_utilisateur}\AppData\Local\4D
- (Windows XP) C:\Documents and Settings\{nom\_utilisateur}\Local Settings\Application Data
- (Mac OS X) /users/{nom\_utilisateur}/library/Caches/4D

Ce dossier n'est pas transporté dans le cadre d'une utilisation déportée du profil utilisateur.

## FIXER PARAMETRE BASE, Lire parametre base

Plusieurs nouveaux sélecteurs (constantes) sont disponibles pour les commandes FIXER PARAMETRE BASE et Lire parametre base.

**Sélecteur = 45** (Enreg requêtes client)

- *Valeurs* : 0 ou de 1 à N (0 = ne pas enregistrer, 1 à N = numéro séquentiel, accolé au nom du fichier).
- *Description* : Démarrage ou arrêt de l'enregistrement des requêtes standard effectuées par le poste client 4D ayant exécuté la commande (hors requêtes Web). Par défaut, la valeur est 0 (pas d'enregistrement des requêtes).

4D vous permet d'enregistrer l'historique des requêtes effectuées par le poste client. Lorsque ce mécanisme est activé, deux fichiers sont créés sur le poste client, dans le sous-dossier *Logs* du dossier local de la base. Il sont nommés "4DRequestsLog\_N" et "4DRequestsLog\_ProcessInfo\_N" où N est le numéro séquentiel de l'historique. Une fois que le fichier *4DRequestsLog* atteint une taille de 10 Mo, il est refermé et un nouveau fichier est généré, avec un numéro séquentiel incrémenté. Si un fichier du même nom existe déjà, il est directement remplacé. Vous pouvez définir le numéro de départ de la séquence à l'aide du paramètre *valeur*.

Ces fichiers texte stockent dans un format tabulé simple diverses informations concernant chaque requête : heure, numéro de

process, taille de la requête, durée de traitement, etc. Ces informations sont particulièrement utiles en phase de mise au point de l'application ou à des fins statistiques.

**Sélecteur = 46** (Chercher par formule serveur)

- *Valeurs* : 0 (utiliser le paramétrage de la base), 1 (exécuter sur le serveur) ou 2 (exécuter sur le client)

- *Description* : Emplacement de l'exécution des commandes CHERCHER PAR FORMULE et CHERCHER PAR FORMULE DANS SELECTION pour la table passée en paramètre.

Dans le cadre de l'exploitation d'une base en client-serveur, les commandes de recherche "par formule" peuvent être exécutées soit sur le serveur soit sur le client :

- dans les bases de données créées avec 4D v11 SQL, ces commandes sont exécutées sur le serveur.

- dans les bases de données converties, ces commandes sont exécutées sur le client, comme dans les versions précédentes de 4D.

- dans les bases de données converties, une préférence spécifique (page Développement/Compatibilité) permet de modifier globalement le lieu d'exécution de ces commandes.

Cette différence de lieu d'exécution influe sur les performances de l'application (l'exécution sur le serveur est généralement plus rapide) mais également sur la programmation. En effet, la valeur des composantes de la formule (notamment les variables appelées via une méthode) diffèrent suivant le contexte d'exécution. Vous pouvez utiliser ce sélecteur pour adapter ponctuellement le fonctionnement de votre application.

Si vous passez 0 dans le paramètre *valeur*, l'emplacement d'exécution des commandes de recherche "par formule" dépendra de la configuration de la base : dans les bases créées avec 4D v11 SQL, les commandes seront exécutées sur le serveur. Dans les bases converties, elles seront exécutées sur le client ou le serveur en fonction des préférences de la base. Passez 1 ou 2 dans *valeur* pour "forcer" l'exécution des commandes respectivement sur le serveur ou sur le client.

- ▶ Cet exemple force temporairement l'exécution sur le client d'une commande de recherche par formule :

```
valCourante:= Lire parametre base([table1];Chercher par formule serveur)  
`Stocker le paramétrage courant  
FIXER PARAMETRE BASE([table1];Chercher par formule serveur;2)
```

```

`Forcer l'exécution sur le client
CHERCHER PAR FORMULE([table1];maformule)
FIXER PARAMETRE BASE([table1];Chercher par formule serveur;
valCourante)

```

`Rétablir le paramétrage courant

**Sélecteur = 47** (Trier par formule serveur)

- *Valeurs* : 0 (utiliser le paramétrage de la base), 1 (exécuter sur le serveur) ou 2 (exécuter sur le client)
- *Description* : Emplacement de l'exécution de la commande TRIER PAR FORMULE pour la table passée en paramètre.  
Dans le cadre de l'exploitation d'une base en client-serveur, la commande TRIER PAR FORMULE peut être exécutée soit sur le serveur soit sur le client. Ce sélecteur permet de définir l'emplacement de l'exécution de cette commande (serveur ou client). Pour plus d'informations, reportez-vous à la description du sélecteur 46, Chercher par formule serveur.

**Sélecteur = 48** (Synchro auto dossier Ressources)

- *Valeurs* : 0 (pas de synchronisation), 1 (synchronisation auto) ou 2 (demander).
- *Description* : Mode de synchronisation dynamique du dossier Ressources du poste client 4D ayant exécuté la commande avec celui du serveur.  
Lorsque le contenu du dossier Ressources sur le serveur a été modifié ou qu'un utilisateur a effectué une demande de synchronisation (par exemple via l'explorateur de ressources), le serveur notifie les clients connectés (cf. [paragraphe "Notification des postes clients", page 26](#)).  
Trois modes de synchronisation sont alors possibles côté client. Le sélecteur Synchro auto dossier Ressources vous permet de définir le mode à utiliser par poste client pour la session courants :
  - 0 (valeur par défaut) = pas de synchronisation dynamique (la demande de synchronisation est ignorée)
  - 1 = synchronisation dynamique automatique
  - 2 = affichage d'une boîte de dialogue sur les postes clients, avec possibilité d'effectuer ou de refuser la synchronisation.
 Le mode de synchronisation peut également être défini globalement dans les Préférences de l'application. Pour plus d'informations, reportez-vous au [paragraphe "Mise à jour des postes clients", page 27](#).

### Sélecteur = 49 (Jointures CHERCHER PAR FORMULE)

- *Valeurs* : 0 (utiliser paramétrages de la base), 1 (toujours utiliser les liens auto) ou 2 (utiliser les jointures SQL si possible).
- *Description* : Mode de fonctionnement des commandes CHERCHER PAR FORMULE et CHERCHER PAR FORMULE DANS SELECTION relatif à l'utilisation de "jointures SQL".

Dans les bases de données créées à compter de la version 11.2 de 4D v11 SQL, ces commandes effectuent des jointures sur le modèle des jointures SQL (cf. [paragraphe "CHERCHER PAR FORMULE, CHERCHER PAR FORMULE DANS SELECTION", page 55](#)). Ce mécanisme permet de modifier la sélection d'une table en fonction d'une recherche effectuée sur une autre table sans que les tables soient reliées par un lien automatique (condition nécessaire dans les versions précédentes de 4D).

Le sélecteur Jointures CHERCHER PAR FORMULE vous permet de définir le mode de fonctionnement des commandes de recherche par formule pour le process courant :

- 0 (valeur par défaut) = utiliser les paramètres courants de la base. Dans les bases de données créées à compter de la version 11.2 de 4D v11 SQL, les "jointures SQL" sont toujours activées pour les recherches par formule. Dans les bases de données converties, ce mécanisme est inactivé par défaut pour des raisons de compatibilité mais peut être mis en oeuvre via une préférence (cf. [paragraphe "Nouvelle préférence de compatibilité", page 56](#))
- 1 = toujours utiliser les liens auto (= fonctionnement des versions précédentes de 4D). Dans ce mode, un lien est nécessaire pour définir la sélection d'une table en fonction de recherches effectuées dans une autre table. 4D n'effectue pas de "jointures SQL".
- 2 = utiliser les jointures SQL si possible (= fonctionnement par défaut des bases créées en version 11.2 et suivantes de 4D v11 SQL ). Dans ce mode, 4D établit des "jointures SQL" pour les recherches par formule lorsque la formule s'y prête (à deux exceptions près, cf. [paragraphe "CHERCHER PAR FORMULE, CHERCHER PAR FORMULE DANS SELECTION", page 55](#)).

# Environnement système

## ENREGISTRER EVENEMENT

ENREGISTRER EVENEMENT (*{typeSortie; }message{; importance}*)

Paramètres	Type	Description
<i>typeSortie</i>	Entier	→ Type de sortie du message
<i>message</i>	Chaîne	→ Contenu du message
<i>importance</i>	Entier	→ Niveau d'importance du message

Les capacités de la commande ENREGISTRER EVENEMENT ont été étendues. Elle peut désormais être utilisée sous Mac OS et un nouveau paramètre facultatif, *typeSortie*, permet de préciser le canal de sortie emprunté par le *message*.

Vous pouvez passer dans *typeSortie* une des constantes suivantes, placées dans le thème "Journal d'événements" :

Constante	Type	Valeur
Vers Observateur Windows	Entier long	0
Vers message débogage	Entier long	1
Vers historique requêtes 4D	Entier long	2
Vers historique commandes 4D	Entier long	3

*Note* Pour refléter les modifications de la commande, le thème de constantes "Journal d'événements Windows" a été renommé "Journal d'événements".

- Vers Observateur Windows  
Le *message* sera envoyé vers l'Observateur d'événements de Windows. Sous Mac OS, la commande ne fait rien (correspond au fonctionnement précédent de la commande).
- Vers message débogage  
Le résultat dépend de la plate-forme :
  - Mac OS : la commande envoie le *message* à la Console
  - Windows : la commande envoie le *message* en tant que message de débogage. Pour pouvoir lire ce message, vous devez disposer de Microsoft Visual Studio ou de l'utilitaire DebugView pour Windows (<http://technet.microsoft.com/en-us/sysinternals/bb896647.aspx>).
- Vers historique requêtes 4D  
La commande inscrit le *message* dans le fichier d'historique des requêtes de 4D, si ce fichier a été activé.

- Vers historique commandes 4D  
La commande inscrit le *message* dans le fichier d'historique des commandes de 4D, si ce fichier a été activé.

---

*Note* Les fichiers d'historique de 4D sont regroupés dans le dossier **Logs**, créé à côté du fichier de structure de la base.

---

## Images

### BLOB VERS IMAGE

BLOB VERS IMAGE(blobImage; image{; codec})

Paramètres	Type	Description
blobImage	BLOB	→ BLOB contenant une image
image	Image	→ Champ ou variable image 4D
codec	Chaîne	→ Identifiant de codec d'image

La commande BLOB VERS IMAGE accepte désormais un troisième paramètre facultatif, *codec*. Vous pouvez passer dans ce paramètre une chaîne précisant le codec à utiliser pour décoder le BLOB.

Si vous passez dans *codec* un codec reconnu par 4D, il est appliqué au BLOB. Vous pouvez obtenir la liste des formats disponibles à l'aide de la commande LISTE CODECS IMAGES.

Si le codec valide correctement le BLOB, 4D retourne l'image et la variable OK prend la valeur 1. Si le codec ne valide pas le BLOB, 4D retourne une image vide et la variable OK prend la valeur 0.

Si vous passez dans *codec* un codec non reconnu par 4D, un nouveau codec est enregistré dynamiquement avec l'identifiant passé en paramètre. 4D retourne une image qui encapsule le BLOB et la variable OK prend la valeur 1. Dans ce cas, pour récupérer le BLOB, il sera nécessaire d'utiliser la commande IMAGE VERS BLOB avec le même identifiant personnalisé. Ce mécanisme particulier permet de répondre à deux besoins spécifiques :

- encapsulation d'un BLOB (qui n'est pas une image) dans une image,
- chargement d'une image sans disposer du codec.

La mise en oeuvre de ces mécanismes permet de notamment de créer des "tableaux de BLOBs" en passant par des tableaux images. Cette technique doit être utilisée avec précaution car, les tableaux étant

entièrement chargés en mémoire, la manipulation de BLOBs de grande taille peut altérer le fonctionnement de l'application.

---

*Note* Un BLOB créé par la commande VARIABLE VERS BLOB est géré automatiquement, il n'est pas nécessaire de passer le codec pour l'encapsuler, le BLOB étant "signé". Pour l'opération inverse dans ce cas, vous devez passer ".4DVarBlob" comme identifiant de codec à la commande IMAGE VERS BLOB.

---

Si vous ne passez pas le paramètre *codec*, comme dans les versions précédentes, 4D tente de déterminer le type de l'image et utilise les codecs disponibles. Si aucun codec ne permet de valider le BLOB, 4D retourne une image vide et la variable OK prend la valeur 0.

## LIRE FICHIER IMAGE LIRE FICHIER IMAGE (nomFichier; image(; \*))

Paramètres	Type	Description
nomFichier	Chaîne	→ Nom ou chemin d'accès complet du fichier à lire, ou chaîne vide
image	Image	← Champ ou variable recevant l'image
*	*	→ <i>Si passé = accepter tout type de fichier</i>

La commande LIRE FICHIER IMAGE accepte désormais un \* en dernier paramètre. Si vous passez ce paramètre, la commande acceptera tout type de fichier. Ce principe permet de manipuler des images sans nécessairement disposer des codecs adéquats (cf. description de la [commande BLOB VERS IMAGE, page 52](#)).

## List box

### LIRE TABLEAUX LISTBOX LISTBOX

LIRE TABLEAUX LISTBOX ({\*; }objet; tabNomsCols; tabNomsEntêtes; tabVarCols; tabVarEntêtes; tabVisibles; *tabStyles*)

Le tableau *tabStyles* peut désormais contenir, pour une list box de type tableau, un quatrième pointeur vers le tableau de type Booléen permettant de définir la visibilité de chaque ligne de la list box.

Pour plus d'informations sur les lignes masquées, reportez-vous au [paragraphe "List box", page 17](#)).

## Outils

### **OUVRIR URL WEB**      `OUVRIR URL WEB (url{; *})`

Les capacités de cette commande ont été étendues dans les domaines suivants :

- prise en charge complète de l'Unicode,
- le paramètre *url* peut désormais contenir aussi bien un URL qu'un chemin d'accès de fichier,
- sous Mac OS, prise en charge de protocoles supplémentaires : (file:, mailto:, news:, http:, etc.)

L'application tentera d'abord d'interpréter le paramètre *url* comme un chemin d'accès de fichier. La commande accepte des ':' sous Mac OS, des '\' sous Windows ou un URL posix commençant par file://. Si c'est le cas, 4D demandera au système d'ouvrir le fichier avec l'application la plus adaptée (par exemple un navigateur pour les .html, Word pour les .doc, etc.). Le paramètre \* est ignoré dans ce cas.

Sinon, l'URL est ouvert en passant par des mécanismes standard.

- ▶ Les exemples suivants permettent d'ouvrir différents types d'URLs :

**OUVRIR URL WEB**( "http://www.4d.com")

**OUVRIR URL WEB**( "file://C:/Users/Laurent/Documents/pending.htm")

**OUVRIR URL WEB**( "C:\Users\Laurent\Documents\pending.htm")

**OUVRIR URL WEB**( "mailto:jean\_martin@4d.fr")

- ▶ Cet exemple permet de lancer une application :

`$fichier:=Selectionner document("";";0)`

**Si** (OK = 1)

**OUVRIR URL WEB**(Document)

**Fin de si**

---

*Note* Pour plus de clarté, la commande **OUVRIR URL WEB**, auparavant placée dans le thème "Serveur Web", appartient désormais au thème "Outils".

---

## Recherches et tris

### CHERCHER PAR TABLEAU DANS SELECTION

CHERCHER PAR TABLEAU DANS SELECTION (champCible; tableau)

Paramètres	Type	Description
champCible	Champ	→ Champ duquel comparer les valeurs
tableau	Tableau	→ Tableau des valeurs recherchées

La nouvelle commande CHERCHER PAR TABLEAU DANS SELECTION recherche dans la table du champ passé en premier paramètre les enregistrements pour lesquels la valeur de *champCible* est égale à au moins une des valeurs des éléments du tableau *tableau*. Les enregistrements trouvés constituent la nouvelle sélection courante.

CHERCHER PAR TABLEAU DANS SELECTION fonctionne de la même manière que CHERCHER PAR TABLEAU. La différence entre ces deux commandes se situe au niveau de la portée de la recherche :

- CHERCHER PAR TABLEAU effectue sa recherche parmi la totalité des enregistrements de la table de *champCible*.
- CHERCHER PAR TABLEAU DANS SELECTION effectue sa recherche uniquement parmi les enregistrements de la sélection courante de la table de *champCible*.

Pour plus d'informations, reportez-vous à la description de la commande CHERCHER PAR TABLEAU.

### CHERCHER PAR FORMULE, CHERCHER PAR FORMULE DANS SELECTION

CHERCHER PAR FORMULE(laTable{; formule})

CHERCHER PAR FORMULE DANS SELECTION(laTable{; formule})

Les commandes CHERCHER PAR FORMULE et CHERCHER PAR FORMULE DANS SELECTION effectuent désormais des "jointures" à la manière du SQL. Avec ce principe, il n'est pas nécessaire qu'un lien automatique structurel existe entre la table A et la table B pour pouvoir utiliser une *formule* contenant [Table\_A]champ\_X = [Table\_B]champ\_Y.

*Note de compatibilité* Ce nouveau mécanisme pouvant générer des dysfonctionnements dans les applications existantes, il est possible de le désactiver. Cette désactivation peut être globale via les Préférences (bases de données converties uniquement) ou par process via les commandes [FIXER PARAMETRE BASE](#), [Lire parametre base](#).

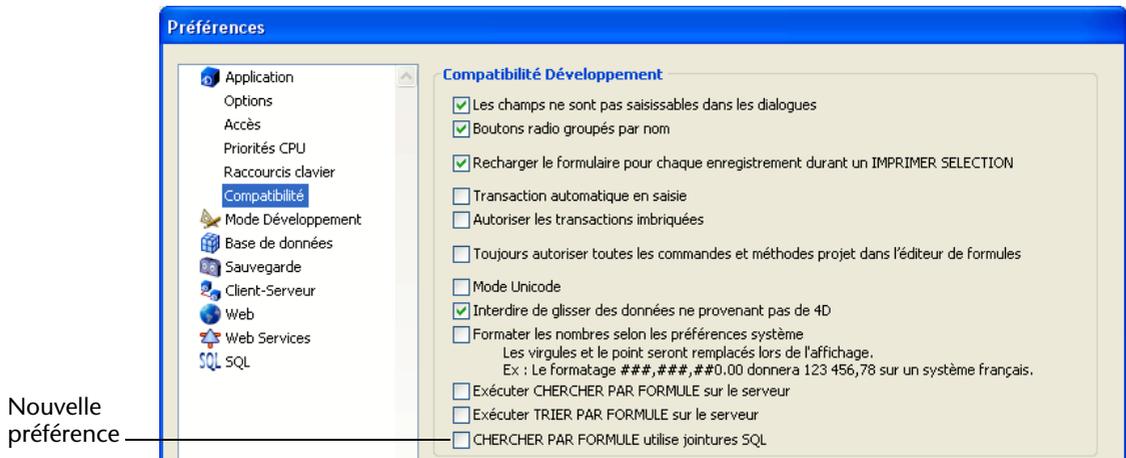
Lorsque le mode "jointure SQL" est activé, les commandes CHERCHER PAR FORMULE et CHERCHER PAR FORMULE DANS SELECTION utilisent toutefois les liens automatiques définis dans l'éditeur de structure dans les cas suivants :

- si la *formule* ne peut se pas décomposer en éléments de la forme {champ ; comparateur ; valeur}
- si deux champs de la même table sont comparés.
- ▶ Cet exemple active les jointures SQL pour une recherche par formule spécifique :

```
$valcourante:= Lire parametre base(Jointures CHERCHER PAR FORMULE)
`Activer les jointures SQL
FIXER PARAMETRE BASE(Jointures CHERCHER PAR FORMULE;2)
`Chercher toutes les lignes de factures du client "ACME" alors que les
`tables ne sont pas liées
CHERCHER PAR FORMULE([ligne_factures] ; [ligne_factures]facture_id =
                        [facture]id & [facture]client = "ACME")
`on rétablit le paramétrage courant
FIXER PARAMETRE BASE(Jointures CHERCHER PAR FORMULE;$valcouran-
te)
```

### Nouvelle préférence de compatibilité

La nouvelle préférence de compatibilité **CHERCHER PAR FORMULE utilise jointures SQL** permet de gérer l'activation et la désactivation du mécanisme de "Jointure SQL" dans les commandes de recherche par formule :



Cette option n'apparaît que dans les bases de données converties en version 11.2 et suivantes. Elle permet l'adaptation progressive des

applications 4D au nouveau mécanisme des recherches par formule. Par défaut, l'option est désélectionnée afin de préserver la compatibilité des traitements. Il est conseillé de l'activer (et de contrôler le code de la base) afin de bénéficier de l'optimisation des commandes de recherche par formule.

---

*Note* Dans les nouvelles bases de données créées à compter de la version 11.2 de 4D v11 SQL, l'option n'apparaît pas, le mécanisme est toujours activé.

---

## Sélections

### VIDER TABLE

VIDER TABLE {(laTable)}

Paramètres	Type	Description
laTable	Table	→ Table de laquelle vous voulez supprimer tous les enregistrements ou Table par défaut si ce paramètre est omis

La commande VIDER TABLE supprime tous les enregistrements de *laTable* de façon très rapide. Si *laTable* est déjà vide, VIDER TABLE ne fait rien.

L'effet de cette commande est semblable à celui d'une séquence TOUT SELECTIONNER / SUPPRIMER SELECTION, toutefois son fonctionnement diffère sur les points suivants :

- Le trigger éventuel n'est pas appelé.
- L'intégrité référentielle des données n'est pas contrôlée.
- Aucune transaction ne doit être en cours dans le process exécutant VIDER TABLE. Si c'est le cas, la commande ne fait rien et la variable système OK prend la valeur 0.
- Si un enregistrement au moins est verrouillé par un autre process, la commande échoue : une erreur est générée et la variable OK prend la valeur 0. L'ensemble système *LockedSet* n'est pas créé.
- Après l'appel de la commande, il n'y a plus de sélection courante ni d'enregistrement courant.
- La variable OK prend la valeur 0 ou 1 selon que la commande a échoué ou réussi.

- L'opération est enregistrée dans le fichier d'historique s'il est présent.

La commande `VIDER TABLE` est donc à manier avec précaution mais est très efficace pour, par exemple, supprimer rapidement des données temporaires.

---

*Note* Le concept et le fonctionnement de cette commande sont proches de ceux de la commande `TRUNCATE (TABLE)` du SQL.

---

## Trigger Sur chargement enregistrement

Le trigger "Sur chargement enregistrement" n'est désormais plus pris en charge dans 4D. Le fonctionnement de ce trigger est incompatible avec les mécanismes internes mis en place dans 4D v11 SQL.

Dans les bases de données converties, ce trigger n'est plus appelé.

A compter de la version 11.2 de 4D, il n'est plus possible de sélectionner ce trigger dans l'Inspecteur de structure. La constante Sur chargement enregistrement a été renommée `_O_Sur chargement enreg` afin de signaler qu'elle est obsolète.

## Web Services (Client)

4D prend désormais en charge l'authentification de services Web en mode DIGEST. Pour cela, deux commandes du thème Web Services (Client) ont été modifiées.

---

*Note* Pour plus d'informations sur la méthode d'authentification DIGEST, reportez-vous à la section "Sécurité des connexions" dans le manuel *Langage* de 4D.

---

### AUTHENTIFIER WEB SERVICE

AUTHENTIFIER WEB SERVICE (nom; motDePasse{; méthodeAuth})

Paramètres	Type		Description
nom	Chaîne	→	Nom d'utilisateur
motDePasse	Chaîne	→	Mot de passe
méthodeAuth	Entier long	→	Méthode d'authentification 0=non définie, 1=BASIC, 2=DIGEST

La commande AUTHENTIFIER WEB SERVICE accepte désormais un troisième paramètre facultatif, *méthodeAuth*. Ce paramètre permet d'indiquer la méthode d'authentification à utiliser pour le prochain appel de la commande APPELER WEB SERVICE.

Vous pouvez passer l'une des valeurs suivantes dans *méthodeAuth* :

- 2 = utiliser la méthode d'authentification DIGEST
- 1 = utiliser la méthode d'authentification BASIC
- 0 (ou paramètre omis) = utiliser la méthode la plus appropriée.

---

*Note* Deux nouvelles options de la commande **FIXER OPTION WEB SERVICE** permettent de contrôler le fonctionnement des requêtes SOAP en mode DIGEST (affichage de la boîte de dialogue d'authentification et conservation des identifiants).

---

## FIXER OPTION WEB SERVICE

FIXER OPTION WEB SERVICE (option; valeur)

La commande FIXER OPTION WEB SERVICE admet deux nouveaux sélecteurs dans le paramètre *option* :

Constante	Type	Valeur
Web Service afficher dial auth	Entier long	4
Web Service effacer infos auth	Entier long	5

Ces sélecteurs permettent de configurer l'authentification des commandes de Web service en mode Digest.

- *option* = Web Service afficher dial auth  
*valeur* = 0 (ne pas afficher le dialogue) ou 1 (afficher le dialogue)  
 Cette option gère l'affichage de boîte de dialogue d'authentification lors de l'exécution de la commande APPELER WEB SERVICE. Par défaut, cette commande ne provoque jamais l'affichage de la boîte de dialogue, vous devez en principe utiliser la commande AUTHENTIFIER WEB SERVICE. Toutefois, si vous souhaitez qu'une boîte de dialogue d'authentification apparaisse pour que l'utilisateur saisisse ses identifiants, vous devez utiliser cette option : passez 1 dans *valeur* pour afficher la boîte de dialogue, et 0 sinon. La boîte de dialogue n'apparaît que si le service Web requiert une authentification.

- *option* = Web Service effacer infos auth  
*valeur* = 0 (ne pas effacer les informations) ou 1 (les effacer)  
 Cette option permet d'indiquer à 4D de mémoriser les informations d'authentification de l'utilisateur (utilisateur, mot de passe, méthode, etc.), dans le but de les réutiliser par la suite. Par défaut, ces informations sont effacées après chaque exécution de la commande APPELER WEB SERVICE. Passez 0 dans *valeur* pour les mémoriser et 1 pour les effacer. A noter que lorsque vous passez 0, les informations sont conservées pendant la session mais ne sont pas stockées.

## Web Services (Serveur)

4D prend désormais en charge l'envoi de structures XML dans le cadre de Web Services.

### DECLARATION SOAP DECLARATION SOAP (variable; type; entrée\_sortie; alias)

La commande DECLARATION SOAP permet désormais de déclarer des variables de type "structure XML" et "référence DOM", aussi bien en entrée qu'en sortie. Lorsque des paramètres de ce type sont définis, aucun traitement ni encodage ne leur est appliqué, les données sont transmises telles quelles.

Pour cela, deux nouvelles constantes peuvent être passées dans le paramètre *type*. Ces constantes sont placées dans le thème "Web Services (Serveur)" :

Constante	Type	Valeur
Est un XML	Entier long	36
Est une référence DOM	Entier long	37

- Paramètres sortants :
  - *Est un XML* indique que le paramètre contient une structure XML,
  - *Est une référence DOM* indique que le paramètre contient la référence DOM d'une structure XML. Dans ce cas, l'insertion de la structure XML dans le message SOAP équivaut à l'exécution de la commande DOM EXPORTER VERS VARIABLE.

---

*Note* Dans le cas de références DOM utilisées en paramètres sortants, il est recommandé d'utiliser des références *globales*, créées par exemple au démarrage et closes à la fermeture de l'application. En effet, une référence DOM créée au sein du Web Service lui-même ne peut pas être

refermée avec DOM FERMER XML sinon le Web Service ne retourne plus rien. Les appels multiples au Web Service impliquent alors la création d'autant de références DOM non refermées, ce qui peut provoquer une saturation de la mémoire.

- Paramètres entrants :
  - *Est un XML* indique que le paramètre doit recevoir un argument XML envoyé par le client SOAP.
  - *Est une référence DOM* indique que le paramètre doit recevoir la référence DOM d'une structure XML correspondant à l'argument XML envoyé par le client SOAP.

**Modification de la WSDL** Ces structures XML seront déclarées par 4D du type "anyType" (indéterminé) dans la WSDL. Si vous souhaitez typer précisément une structure XML, vous devez sauvegarder le fichier WSDL et ajouter manuellement le schéma de données souhaité dans la section <types> de la WSDL.

- ▶ Cet exemple illustre le résultat des différents types de déclarations :

**TOUT SELECTIONNER**([Contact])

`Construction d'une structure XML à partir de la sélection de Contacts  
 `et stockage du XML dans un BLOB

**C\_BLOB**(ws\_vx\_xmlBlob)

*getContactsXML* (->ws\_vx\_xmlBlob)

`Récupération de la structure XML dans une variable texte

**C\_TEXTE**(ws\_vt\_xml)

ws\_vt\_xml:=**BLOB vers texte**(ws\_vx\_xmlBlob;UTF8 Texte sans longueur)

`Récupération d'une référence DOM vers la structure XML

**C\_TEXTE**(ws\_vt\_refXML)

ws\_vt\_refXML:=**DOM Analyser variable XML**(ws\_vt\_xml)

`Test des différentes déclarations

**DECLARATION SOAP**(ws\_vx\_xmlBlob;Est un BLOB ;SOAP Sortie ;  
 "contactListsX")

`Le XML est converti en Base64 par 4D

**DECLARATION SOAP**(ws\_vt\_xml;Est un Texte;SOAP Sortie;  
 "contactListsText")

`Le XML est converti en texte par 4D (< > deviennent des entités)

`Nouveautés v 11.2

**DECLARATION SOAP**(ws\_vt\_xml; Est un XML; SOAP Sortie; "contactsXML")

`Le XML est passé en texte XML

**DECLARATION SOAP**(ws\_vx\_xmlBlob; Est un XML; SOAP Sortie; "contactsBlob")

`Le XML est passé en BLOB XML

**DECLARATION SOAP**(ws\_vt\_refXML; Est une référence DOM; SOAP Sortie; "contactByRef")

`Le XML est passé en tant que référence

## Zone Web

Ce nouveau thème de commandes permet de gérer les nouveaux objets de type "Zone Web" apparus dans la version 11.2 de 4D v11 SQL. Les objets "Zone Web" sont détaillés dans le [paragraphe "Zones Web", page 28](#).

### WA OUVRIR URL

WA OUVRIR URL({\*;}objet; url)

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
url	Chaîne	→ URL à charger dans la zone Web

La commande WA OUVRIR URL charge dans la zone Web désignée par les paramètres \* et *objet* l'URL passé dans le paramètre *url*.

Si une chaîne vide est passée dans *url*, la commande WA OUVRIR URL ne fait rien et aucune erreur n'est générée. Pour charger une page vide dans la zone Web, passez la chaîne "about:blank" dans *url*.

Comme la commande existante [OUVRIR URL WEB](#), WA OUVRIR URL accepte plusieurs types de syntaxes dans le paramètre *url* pour désigner les fichiers :

- syntaxe posix : "file://c:/Mon%20Fichier"
- syntaxe système : "c:\MonDossier\MonFichier" (Windows) ou "MonDisque:MonDossier:MonFichier" (Mac OS).

Cette commande a le même effet que la modification de la valeur de la variable "URL" associée à la zone. Par exemple, si la variable de la zone est nommée *MaZoneW\_url* :

```
MaZoneW_url="http://www.4d.com/"
```

équivalent à :

```
WA OUVRIER URL(MaZoneW;"http://www.4d.com/")
```

## WA Lire URL courant **WA Lire URL courant** ({\*;}objet) → Chaîne

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formule	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
Résultat	Chaîne	← URL actuellement chargé dans la zone Web

La commande WA Lire URL courant retourne l'adresse URL de la page affichée dans la zone Web désignée par les paramètres \* et *objet*.

Si l'URL courant n'est pas disponible, la commande retourne une chaîne vide.

Si la page Web est entièrement chargée, la valeur retournée par la fonction est identique à celle de la variable "URL" associée à la zone Web. Si la page est en cours de chargement, les deux valeurs seront différentes : la fonction retourne l'URL entièrement chargé et la variable contient l'URL en cours de chargement.

- ▶ La page affichée est l'URL "www.apple.com" et la page "www.4d.com" est en cours de chargement :

```
$url:=WA Lire URL courant(MaZoneW) `retourne "http://www.apple.com"
`La variable URL associée contient "http://www.4d.com"
```

## WA OUVRIR URL PRECEDENT

WA OUVRIR URL PRECEDENT({\*;}objet)

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)

La commande WA OUVRIR URL PRECEDENT charge dans la zone Web désignée par les paramètres \* et *objet* l'URL précédent dans la séquence d'URLs ouverts.

S'il n'y a pas d'URL précédent, la commande ne fait rien. Vous pouvez tester la disponibilité d'un URL précédent à l'aide de la commande WA URL precedent disponible.

## WA URL precedent disponible

WA URL precedent disponible ({\*;}objet) → Booléen

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
Résultat	Booléen	← Vrai s'il existe un URL précédent dans la séquence d'URLs ouverts, Faux sinon

La commande WA URL precedent disponible permet de savoir s'il existe un URL précédent disponible dans la séquence d'URLs ouverts dans la zone Web désignée par les paramètres \* et *objet*.

La commande retourne Vrai si un URL existe et Faux sinon. Cette commande permet notamment, dans la cadre d'une interface personnalisée, d'activer ou d'inactiver des boutons de navigation.

## WA OUVRIR URL SUIVANT

WA OUVRIR URL SUIVANT({\*;}objet)

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)

La commande WA OUVRIR URL SUIVANT charge dans la zone Web désignée par les paramètres \* et *objet* l'URL suivant dans la séquence d'URLs ouverts.

S'il n'y a pas d'URL suivant (c'est-à-dire si l'utilisateur n'a jamais effectué de retour à l'URL précédent), la commande ne fait rien. Vous pouvez tester la disponibilité d'un URL suivant à l'aide de la commande WA URL suivant disponible.

## WA URL suivant disponible

WA URL suivant disponible ({\*;}objet) → Booléen

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
Résultat	Booléen	← Vrai s'il existe un URL suivant dans la séquence d'URLs ouverts, Faux sinon

La commande WA URL suivant disponible permet de savoir s'il existe un URL suivant disponible dans la séquence d'URLs ouverts dans la zone Web désignée par les paramètres \* et *objet*.

La commande retourne Vrai si un URL existe et Faux sinon. Cette commande permet notamment, dans la cadre d'une interface personnalisée, d'activer ou d'inactiver des boutons de navigation.

## WA ACTUALISER URL

WA ACTUALISER URL ({\*;}objet)

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)

La commande WA ACTUALISER URL provoque le rechargement de l'URL courant affiché dans la zone Web désignée par les paramètres \* et *objet*.

## WA ARRETER CHARGEMENT URL

WA ARRETER CHARGEMENT URL ({\*;}objet)

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)

La commande WA ARRETER CHARGEMENT URL stoppe le chargement des ressources de l'URL courant de la zone Web désignée par les paramètres \* et *objet*.

## WA Executer JavaScript

WA Executer JavaScript ({\*;}objet; codeJS) → Chaîne

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
codeJS	Chaîne	→ Code JavaScript
Résultat	Chaîne	← Résultat de l'exécution

La commande WA Executer JavaScript exécute dans la zone Web désignée par les paramètres \* et *objet* le code JavaScript passé dans *codeJS*.

Sous Mac OS, la commande retourne le résultat.

Sous Windows, la commande retourne une chaîne vide. Utilisez la commande WA EXECUTER FONCTION JAVASCRIPT.

## WA EXECUTER FONCTION JAVASCRIPT

### ► Exemple

```
$résultat:=WA Executer JavaScript(MaZoneW;"history.back()")
```

WA EXECUTER FONCTION JAVASCRIPT ({\*;}objet; fonctionJS; résultat|\*{;  
param1;...;paramN})

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
fonctionJS	Chaîne	→ Nom de la fonction JavaScript à exécuter
résultat *	Variable	→ Résultat de la fonction (si attendu) ← ou * pour une fonction sans résultat
param	Chaîne	→ Paramètre(s) à passer à la fonction

La commande WA EXECUTER FONCTION JAVASCRIPT exécute dans la zone Web désignée par les paramètres \* et *objet* la fonction JavaScript *fonctionJS* et retourne facultativement son résultat dans le paramètre *résultat*.

Si la fonction ne retourne pas de résultat, passez \* dans le paramètre *résultat*.

Vous pouvez passer dans *param* une ou plusieurs chaîne(s) contenant les paramètres de la fonction.

### ► Appel d'une fonction JavaScript avec 3 paramètres :

```
$JavaScriptFunction:="TheFunctionToBeExecuted"  
$Param1:="10"  
$Param2:="true"  
$Param3:="1,000.2" `notez "," comme séparateur de milliers et "." comme  
séparateur décimal
```

```
WA EXECUTER FONCTION JAVASCRIPT(MaZoneW; $JavaScriptFunction;  
$Result; $Param1; $Param2; $Param3)
```

## WA FIXER FILTRES URL

WA FIXER FILTRES URL({\*;}objet; tabFiltres; tabAutorisRefus)

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
tabFiltres	Tableau chaîne	→ Tableau de filtres
tabAutorisRefus	Tableau Booléen	→ Tableau autoriser-refuser

La commande WA FIXER FILTRES URL permet de mettre en place un ou plusieurs filtre(s) pour la zone Web désignée par les paramètres \* et *objet*.

Avant le chargement de toute page demandé par l'utilisateur, 4D consulte la liste des filtres afin de vérifier si l'URL cible est autorisé ou non. L'évaluation de l'URL est basée sur le contenu des tableaux *tabFiltres* et *tabAutorisRefus*.

Si l'URL demandé n'est pas autorisé, il n'est pas chargé et l'événement formulaire Sur filtrage URL est généré (cf. [paragraphe "Evénements formulaire", page 33](#)).

Les tableaux *tabFiltres* et *tabAutorisRefus* doivent être synchronisés.

- Chaque ligne du tableau *tabFiltres* doit contenir un URL devant être filtré. Vous pouvez utiliser le \* comme joker de remplacement pour un ou plusieurs caractère(s).
- Chaque ligne correspondante dans le tableau *tabAutorisRefus* doit contenir un booléen indiquant si l'URL doit être autorisé (Vrai) ou refusé (Faux).

En cas de contradiction au niveau des paramétrages (autorisation et refus d'un même URL), le paramétrage pris en compte est le dernier.

Pour désactiver le filtrage des URLs, appelez la commande en lui passant des tableaux vides ou en passant respectivement les valeurs "\*" et Vrai dans la dernière ligne des tableaux *tabFiltres* et *tabAutorisRefus*.

Une fois la commande exécutée, les filtres deviennent une propriété de la zone Web. Si les tableaux *tabFiltres* et *tabAutorisRefus* sont supprimés ou réinitialisés, les filtres restent actifs tant que la commande n'a pas

été exécutée à nouveau. Pour connaître les filtres actifs pour une zone, vous devez utiliser la commande [WA LIRE FILTRES URL](#).

**Important** : Le filtrage des URLs effectué par cette commande s'applique uniquement à la variable "URL" associée à la zone Web (variable généralement saisissable et affichée dans le formulaire). Le filtrage ne s'applique pas à la commande [WA OUVRIR URL](#) ni aux autres commandes de navigation.

- ▶ Vous souhaitez interdire l'accès à tous les sites web .org, .net et .fr.

```
TABLEAU TEXTE($filters;0)
TABLEAU BOOLEEN($AllowDeny;0)
AJOUTER A TABLEAU($filters;"*.org")
AJOUTER A TABLEAU($AllowDeny;Faux)
AJOUTER A TABLEAU($filters;"*.net")
AJOUTER A TABLEAU($AllowDeny;Faux)
AJOUTER A TABLEAU($filters;"*.fr")
AJOUTER A TABLEAU($AllowDeny;Faux)
WA FIXER FILTRES URL(MaZoneW;$filters;$AllowDeny)
```

- ▶ Vous souhaitez interdire l'accès à tous les sites web sauf les sites russes (.ru) :

```
TABLEAU TEXTE($filters;0)
TABLEAU BOOLEEN($AllowDeny;0)

AJOUTER A TABLEAU($filters;"*") `Tout sélectionner
AJOUTER A TABLEAU($AllowDeny;Faux) `Tout interdire
AJOUTER A TABLEAU($filters;"www*.ru") `Sélectionner *.ru
AJOUTER A TABLEAU($AllowDeny;Vrai) `Autoriser
WA FIXER FILTRES URL(MaZoneW;$filters;$AllowDeny)
```

- ▶ Vous souhaitez donner accès aux sites Web 4D uniquement (.com, .fr, .es, etc.) :

```
TABLEAU TEXTE($filters;0)
TABLEAU BOOLEEN($AllowDeny;0)

AJOUTER A TABLEAU($filters;"*") `Tout sélectionner
AJOUTER A TABLEAU($AllowDeny;Faux) `Tout interdire
AJOUTER A TABLEAU($filters;"www.4D.*") `Sélectionner 4d.fr, 4d.com...
AJOUTER A TABLEAU($AllowDeny;Vrai) `Autoriser
WA FIXER FILTRES URL(MaZoneW;$filters;$AllowDeny)
```

- ▶ Vous souhaitez autoriser l'accès local à la documentation uniquement (située dans le dossier C://doc) :

```
TABLEAU TEXTE($filters;0)
TABLEAU BOOLEEN($AllowDeny;0)

AJOUTER A TABLEAU($filters;"*") `Tout sélectionner
AJOUTER A TABLEAU($AllowDeny;Faux) `Tout interdire
AJOUTER A TABLEAU($filters;"file://C:/doc/*")
    `Sélectionner le chemin file:// autorisé
AJOUTER A TABLEAU($AllowDeny;Vrai)-> Autoriser
WA FIXER FILTRES URL(MaZoneW;$filters;$AllowDeny)
```

- ▶ Vous souhaitez autoriser tous les sites sauf un, par exemple celui d'Elcaro :

```
TABLEAU TEXTE($filters;0)
TABLEAU BOOLEEN($AllowDeny;0)

AJOUTER A TABLEAU($filters;"*")
AJOUTER A TABLEAU($AllowDeny;Vrai) `Tout autoriser
AJOUTER A TABLEAU($filters;"*elcaro*") `Interdire tout ce qui contient el-
caro
AJOUTER A TABLEAU($AllowDeny;Faux)
WA FIXER FILTRES URL(MaZoneW;$filters;$AllowDeny)
```

- ▶ Vous souhaitez interdire des adresses IP spécifiques :

```
TABLEAU TEXTE($filters;0)
TABLEAU BOOLEEN($AllowDeny;0)

AJOUTER A TABLEAU($filters;"*") `Tout sélectionner
AJOUTER A TABLEAU($AllowDeny;Vrai) `Tout autoriser
AJOUTER A TABLEAU($filters;"86.83.*") `Sélectionner les IP débutant par
86.83.
AJOUTER A TABLEAU($AllowDeny;Faux) `Interdire
AJOUTER A TABLEAU($filters;"86.1*") `Sélectionner les IP débutant par
86.1 (86.10, 86.135 etc.)
AJOUTER A TABLEAU($AllowDeny;Faux) `Interdire
WA FIXER FILTRES URL(MaZoneW;$filters;$AllowDeny)
```

(A noter que l'adresse IP d'un domaine peut varier).

*Référence :* [WA LIRE FILTRES URL](#)



## WA LIRE FILTRES URL

WA LIRE FILTRES URL({\*};objet; tabFiltres; tabAutorisRefus)

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
tabFiltres	Tableau chaîne	← Tableau de filtres
tabAutorisRefus	Tableau Booléen	← Tableau autoriser-refuser

La commande WA LIRE FILTRES URL retourne dans les tableaux *tabFiltres* et *tabAutorisRefus* les filtres actifs dans la zone Web désignée par les paramètres \* et *objet*. Si aucun filtre n'est actif, les tableaux sont retournés vides.

Les filtres sont installés par la commande WA FIXER FILTRES URL. Si les tableaux ont été réinitialisés au cours de la session, la commande WA LIRE FILTRES URL vous permet de connaître le paramétrage courant.

**Référence :** [WA FIXER FILTRES URL](#)

## WA FIXER FILTRES LIENS EXTERNES

WA FIXER FILTRES LIENS EXTERNES({\*};objet; tabFiltres; tabAutorisRefus)

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
tabFiltres	Tableau chaîne	→ Tableau de filtres
tabAutorisRefus	Tableau Booléen	→ Tableau autoriser-refuser

La commande WA FIXER FILTRES LIENS EXTERNES permet de mettre en place un ou plusieurs filtre(s) de liens externes pour la zone Web désignée par les paramètres \* et *objet*. Les filtres de liens externes déterminent si un URL associé à la page courante via un lien doit être ouvert dans la zone Web ou dans le navigateur Web par défaut de la machine.

Lorsque l'utilisateur clique sur un lien dans la page courante, 4D consulte la liste des filtres de liens externes afin de vérifier si l'URL demandé doit être ouvert dans le navigateur de la machine. Si c'est le cas, la page correspondant à l'URL est affichée dans le navigateur Web et l'événement formulaire Sur ouverture lien externe est généré (cf. [paragraphe "Événements formulaire", page 33](#)). Sinon (fonctionnement par défaut), la page correspondant à l'URL est affichée dans la zone Web. L'évaluation de l'URL est basée sur le contenu des tableaux *tabFiltres* et *tabAutorisRefus*.

Les tableaux *tabFiltres* et *tabAutorisRefus* doivent être synchronisés.

- Chaque ligne du tableau *tabFiltres* doit contenir un URL devant être filtré. Vous pouvez utiliser le \* comme joker de remplacement pour un ou plusieurs caractère(s).
- Chaque ligne correspondante dans le tableau *tabAutorisRefus* doit contenir un booléen indiquant si l'URL doit être ouvert dans la zone Web (Vrai) ou dans le navigateur Web (Faux).

En cas de contradiction au niveau des paramètres (autorisation et refus d'un même URL), le paramétrage pris en compte est le dernier.

Pour désactiver le filtrage des URLs, appelez la commande en lui passant des tableaux vides ou en passant respectivement les valeurs "\*" et Vrai dans la dernière ligne des tableaux *tabFiltres* et *tabAutorisRefus*.

**Important :** Le filtrage établi par la commande [WA FIXER FILTRES URL](#) est pris en compte avant celui de [WA FIXER FILTRES LIENS EXTERNES](#). Cela signifie que si un URL est refusé à cause d'un filtre de la commande [WA FIXER FILTRES URL](#), il ne pourra pas être ouvert dans un navigateur même s'il est explicitement défini par la commande [WA FIXER FILTRES LIENS EXTERNES](#) (cf. exemple 2).

- ▶ Cet exemple provoquera l'ouverture de sites dans des navigateurs externes :

```
TABLEAU ALPHA(0;$filters;0)
TABLEAU BOOLEEN($AllowDeny;0)
```

```
AJOUTER A TABLEAU($filters;"*www.google.*") `Sélectionner "google"
AJOUTER A TABLEAU($AllowDeny;Faux)
```

    `Faux : ce lien sera ouvert dans un navigateur externe

```
AJOUTER A TABLEAU($filters;"*www.apple.*")
AJOUTER A TABLEAU($AllowDeny;Faux)
```

`Faux : ce lien sera ouvert dans un navigateur externe  
**WA FIXER FILTRES LIENS EXTERNES**(MaZoneW;\$filters;\$AllowDeny)

- ▶ Cet exemple combine des filtrages de sites et de liens externes :

```
TABLEAU ALPHA(0;$filters;0)
TABLEAU BOOLEEN($AllowDeny;0)
AJOUTER A TABLEAU($filters;"*www.google.*") `Sélectionner "google"
AJOUTER A TABLEAU($AllowDeny;Faux) `Interdire ce lien
WA FIXER FILTRES URL(MaZoneW;$filters;$AllowDeny)
```

```
TABLEAU ALPHA(0;$filters;0)
TABLEAU BOOLEEN($AllowDeny;0)
AJOUTER A TABLEAU($filters;"*www.google.*") `Sélectionner "google"
AJOUTER A TABLEAU($AllowDeny;Faux)
`Faux : ce lien devrait être ouvert dans un navigateur externe, mais ce
`paramétrage est sans effet car le lien sera bloqué du fait du filtrage d'URL.
WA FIXER FILTRES LIENS EXTERNES(MaZoneW;$filters;$AllowDeny)
```

## WA LIRE FILTRES LIENS EXTERNES

**WA LIRE FILTRES LIENS EXTERNES**({\*;}objet; tabFiltres; tabAutorisRefus)

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
tabFiltres	Tableau chaîne	← Tableau de filtres
tabAutorisRefus	Tableau Booléen	← Tableau autoriser-refuser

La commande **WA LIRE FILTRES LIENS EXTERNES** retourne dans les tableaux *tabFiltres* et *tabAutorisRefus* les filtres de liens externes de la zone Web désignée par les paramètres \* et *objet*. Si aucun filtre n'est actif, les tableaux sont retournés vides.

Les filtres sont installés par la commande **WA FIXER FILTRES LIENS EXTERNES**. Si les tableaux ont été réinitialisés au cours de la session, la commande **WA LIRE FILTRES LIENS EXTERNES** vous permet de connaître le paramétrage courant.

**Référence** : **WA FIXER FILTRES LIENS EXTERNES**

## WA LIRE DERNIERE ERREUR URL

WA LIRE DERNIERE ERREUR URL ({\*;}objet; url; description; codeErreur)

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
url	Chaîne	← URL à l'origine de l'erreur
description	Chaîne	← Description de l'erreur (Mac OS)
codeErreur	Entier long	← Code de l'erreur

La commande WA LIRE DERNIERE ERREUR URL vous permet de récupérer plusieurs informations relatives à la dernière erreur ayant eu lieu dans la zone Web désignée par les paramètres \* et *objet*.

Ces informations sont retournées dans trois variables :

- *url* : l'URL ayant provoqué l'erreur.
- *description* (Mac OS uniquement) : un texte décrivant l'erreur (si disponible). S'il n'est pas possible d'associer un texte à l'erreur, une chaîne vide est retournée. Sous Windows, ce paramètre est toujours retourné vide.
- *codeErreur* : code de l'erreur.
  - Si le code est  $\geq 400$ , il s'agit d'une erreur liée au protocole HTTP. Pour plus d'informations sur ce type d'erreur, reportez-vous à l'adresse : <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>
  - Sinon, il s'agit d'une erreur retournée par le WebKit (Mac OS) ou ActiveX (Windows).

Il est judicieux d'appeler cette commande dans le cadre de l'événement formulaire Sur erreur chargement URL afin de connaître la cause de l'erreur qui vient de se produire. Pour plus d'informations, reportez-vous au [paragraphe "Événements formulaire", page 33](#).

## WA Lire dernier URL filtre

WA Lire dernier URL filtre ({\*;}objet) → Chaîne

Paramètres	Type		Description
*	*	→	Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→	Nom d'objet (si * est spécifié) ou Variable (si * est omis)
Résultat	Chaîne	←	Dernier URL filtré

La commande WA Lire dernier URL filtre retourne le dernier URL ayant été filtré dans la zone Web désignée par les paramètres \* et *objet*.

L'URL peut avoir été filtré pour l'une des raisons suivantes :

- L'URL est interdit à cause d'un filtre (commande [WA FIXER FILTRES URL](#)),
- le lien est ouvert dans le navigateur par défaut (commande [WA FIXER FILTRES LIENS EXTERNES](#)),
- l'URL tentait d'ouvrir une fenêtre pop up.

Il est judicieux d'appeler cette commande dans le contexte des événements formulaire Sur filtrage URL, Sur ouverture lien externe et Sur refus ouverture fenêtre afin de connaître l'URL filtré. Pour plus d'informations, reportez-vous au [paragraphe "Événements formulaire"](#), page 33.

## WA LIRE HISTORIQUE URL

WA LIRE HISTORIQUE URL({\*;}objet; tabUrls{; direction{; tabTitres{}})

Paramètres	Type		Description
*	*	→	Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→	Nom d'objet (si * est spécifié) ou Variable (si * est omis)
tabUrls	Tab Alpha	←	Tableau des URLs visités
direction	Entier	→	0 ou omis=Liste des URLs précédents, 1=Liste des URLs suivants
tabTitres	Tab Alpha	←	Tableau des titres de fenêtres

La commande WA LIRE HISTORIQUE URL retourne un ou deux tableaux contenant les URLs visités au cours de la session dans la zone Web désignée par les paramètres \* et *objet*. Elle permet de construire une interface de navigation personnalisée.

Les informations fournies concernent la session, c'est-à-dire la navigation effectuée dans une même zone Web tant que le formulaire n'a pas été refermé.

Le tableau *tabUrls* est rempli avec la liste des URLs visités. En fonction de la valeur du paramètre *direction* (s'il est passé), le tableau récupère la liste des URLs précédents (fonctionnement par défaut) ou la liste des URLs suivants. Ces listes correspondent au contenu des boutons standard Précédent et Suivant des navigateurs.

Les URLs sont classés par ordre chronologique.

Passez dans *direction* une valeur indiquant la liste à récupérer. Vous pouvez utiliser l'une des constantes suivantes, placées dans le thème "Zone Web" :

Constante	Type	Valeur
wa URLs précédents	Entier long	0
wa URLs suivants	Entier long	1

Si vous omettez le paramètre *direction*, la valeur 0 est utilisée.

S'il est passé, le paramètre *tabTitres* contient la liste des noms de fenêtres associés aux URLs. Ce tableau est synchronisé avec le tableau *tabUrls*.

## WA Créer menu historique URL

WA Créer menu historique URL ({\*;}objet{; direction}) → RefMenu

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
direction	Entier	→ 0 ou omis=Liste des URLs précédents, 1=Liste des URLs suivants
Résultat	RefMenu	← Référence du menu

La commande WA Créer menu historique URL crée et remplit un menu pouvant être utilisé directement pour la navigation parmi les URLs visités au cours de la session dans la zone Web désignée par les paramètres \* et *objet*. Elle permet de construire une interface de navigation personnalisée.

Les informations fournies concernent la session, c'est-à-dire la navigation effectuée dans une même zone Web tant que le formulaire n'a pas été refermé.

Passez dans *direction* une valeur indiquant la liste à récupérer. Vous pouvez utiliser l'une des constantes suivantes, placées dans le thème "Zone Web" :

Constante	Type	Valeur
wa URLs précédents	Entier long	0
wa URLs suivants	Entier long	1

Si vous omettez le paramètre *direction*, la valeur 0 est utilisée.

Une fois le menu généré, vous pouvez l'afficher via la commande de 4D Pop up menu dynamique et le manipuler via les commandes standard de gestion des menus de 4D. La référence *RefLigne* (de type chaîne) retournée par la commande Pop up menu dynamique contient l'URL de la page visitée (voir exemple).

Appelez la commande EFFACER MENU pour supprimer un menu historique d'URL lorsqu'il est devenu inutile.

- Le code suivant pourrait être associé à un bouton 3D avec pop up menu libellé "Précédent" :

#### Au cas ou

  `Clic simple

: (Evenement formulaire=Sur clic)

**WA OUVRIR URL PRECEDENT**(WA\_zone)

  `Clic sur la flèche -> affichage du pop up

: (Evenement formulaire=Sur clic flèche)

  `Créer un menu historique précédent

  \$Menu:=**WA Créer menu historique URL**(WA\_zone;

wa URLs précédents)

  `Afficher ce menu dans un pop up

  \$URL:=**Pop up menu dynamique**(\$Menu)

  `Si une ligne est sélectionnée

**Si** (\$URL#"")

**WA OUVRIR URL**(WA\_zone;\$URL) `Ouvrir la page Web

**Fin de si**

  `Effacer le menu pour libérer la mémoire

**EFFACER MENU**(\$Menu)

#### Fin de cas

**Référence** : Pop up menu dynamique, EFFACER MENU

## WA Lire titre page

WA Lire titre page ({\*;}objet) → Chaîne

Paramètres	Type		Description
*	*	→	Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→	Nom d'objet (si * est spécifié) ou Variable (si * est omis)
Résultat	Chaîne	←	Titre de la page courante

La commande WA Lire titre page retourne le titre de la page courante ou en cours d'affichage dans la zone Web désignée par les paramètres \* et *objet*. Le titre correspond à la balise HTML "Title".

Cette commande retourne une chaîne vide s'il n'y a pas de titre disponible à l'URL courant.

## WA FIXER CONTENU PAGE

WA FIXER CONTENU PAGE({\*;}objet; contenu; baseURL)

Paramètres	Type		Description
*	*	→	Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→	Nom d'objet (si * est spécifié) ou Variable (si * est omis)
contenu	Chaîne	→	Code HTML source
baseURL	Chaîne	→	URL pour les références relatives (Mac OS)

La commande WA FIXER CONTENU PAGE remplace la page affichée dans la zone Web désignée par les paramètres \* et *objet* par le code HTML passé dans le paramètre *contenu*.

Le paramètre *baseURL* permet de définir sous Mac OS un URL de base qui sera ajouté devant les liens relatifs éventuellement présents dans la page. Sous Windows, ce paramètre est sans effet, l'URL de base est indéfini. Il n'est donc pas possible d'utiliser des références relatives sur cette plate-forme.

---

*Note* Sous Windows, il est impératif qu'une page ait déjà été chargée dans la zone Web avant que cette commande puisse être appelée. Si nécessaire, vous pouvez passer l'URL "about:blank" afin de charger une page blanche.

---

- Affichage de la phrase "Hello world !" et définition d'un URL de base "file:/// " (Mac OS uniquement) :

**WA FIXER CONTENU PAGE**(MaZoneW;"<html><body><h1>Hello World!</h1></body></html>";"file:///")

## WA Lire contenu page

WA Lire contenu page({\*};objet) → Chaîne

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
Résultat	Chaîne	← Code HTML source

La commande WA Lire contenu page retourne le code HTML de la page courante ou en cours d'affichage dans la zone Web désignée par les paramètres \* et *objet*.

Cette commande retourne une chaîne vide si le contenu de la page courante n'est pas disponible.

## WA AGRANDIR TEXTE PAGE

WA AGRANDIR TEXTE PAGE({\*};objet)

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)

La commande WA AGRANDIR TEXTE PAGE augmente la taille du texte affiché dans la zone Web désignée par les paramètres \* et *objet*.

Sous Mac OS, la portée de cette commande est la session 4D : le paramétrage effectué par cette commande n'est pas conservé après la fermeture de l'application 4D.

Sous Windows, la portée de cette commande est globale : le paramétrage est conservé après la fermeture de l'application 4D.

## WA REDUIRE TEXTE PAGE

WA REDUIRE TEXTE PAGE({\*;}objet)

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)

La commande WA AGRANDIR TEXTE PAGE réduit la taille du texte affiché dans la zone Web désignée par les paramètres \* et *objet*.

Sous Mac OS, la portée de cette commande est la session 4D : le paramétrage effectué par cette commande n'est pas conservé après la fermeture de l'application 4D.

Sous Windows, la portée de cette commande est globale : le paramétrage est conservé après la fermeture de l'application 4D.

## WA FIXER PREFERENCE

WA FIXER PREFERENCE({\*;}objet; sélecteur; valeur)

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
sélecteur	Entier long	→ Préférence à modifier
valeur	Booléen	→ Valeur de la préférence (Vrai = autorisé, Faux = non autorisé)

La commande WA FIXER PREFERENCE permet de fixer différentes préférences pour la zone Web désignée par les paramètres \* et *objet*.

Passez dans le paramètre *sélecteur* la préférence à modifier et dans *valeur* la valeur à lui attribuer. Vous pouvez passer dans *sélecteur* l'une des constantes suivantes, placées dans le thème "Zone Web" :

Constante	Type	Valeur
wa autoriser applets Java	Entier long	1
wa autoriser JavaScript	Entier long	2
wa autoriser plugins	Entier long	3
wa autoriser menu contextuel	Entier long	4



Pour chaque préférence, passez *Vrai* dans *valeur* pour l'activer et *Faux* pour l'inactiver.

Voici la signification des sélecteurs :

- *wa autoriser applets Java* : permet d'autoriser l'exécution d'applets Java dans la zone Web.
- *wa autoriser JavaScript* : permet d'autoriser l'exécution de code JavaScript dans la zone Web.
- *wa autoriser plugins* : permet d'autoriser l'installation de plug-ins dans la zone Web.
- *wa autoriser menu contextuel* : permet d'autoriser l'affichage du menu contextuel standard dans la zone Web.

**Référence** : [WA LIRE PREFERENCE](#)

## WA LIRE PREFERENCE

WA LIRE PREFERENCE({\*;}objet; sélecteur; valeur)

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
sélecteur	Entier long	→ Préférence à lire
valeur	Variable	← Valeur courante de la préférence

La commande WA LIRE PREFERENCE permet de lire la valeur courante d'une préférence dans la zone Web désignée par les paramètres \* et *objet*.

Passez dans le paramètre *sélecteur* la préférence à lire. Vous pouvez passer l'une des constantes suivantes, placées dans le thème "Zone Web" :

Constante	Type	Valeur
<i>wa autoriser applets Java</i>	Entier long	1
<i>wa autoriser JavaScript</i>	Entier long	2
<i>wa autoriser plugins</i>	Entier long	3
<i>wa autoriser menu contextuel</i>	Entier long	4

Pour plus d'informations sur ces préférences, reportez-vous à la description de la [commande WA FIXER PREFERENCE](#), page 80.

Passez dans le paramètre *valeur* une variable devant recevoir la valeur courante de la préférence. Le type de la variable dépend de la préférence. Dans la version actuelle de 4D v11 SQL, la variable *valeur* est toujours de type booléen : elle contient Vrai si la préférence est active et Faux sinon.

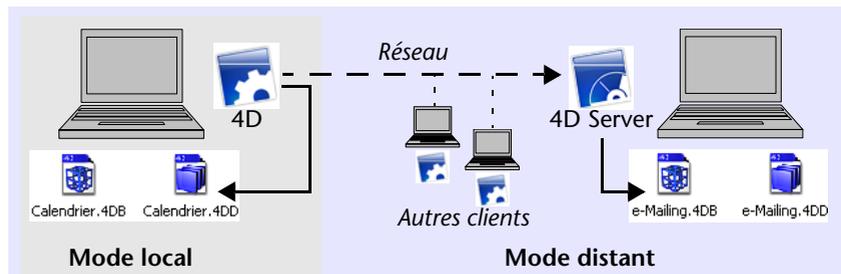
# 4

## 4D Server v11 SQL

Bienvenue dans 4D Server v11 SQL. Cette nouvelle version de 4D Server propose de nombreuses évolutions majeures vous permettant de tirer parti de la puissance de la gamme 4D v11 SQL.

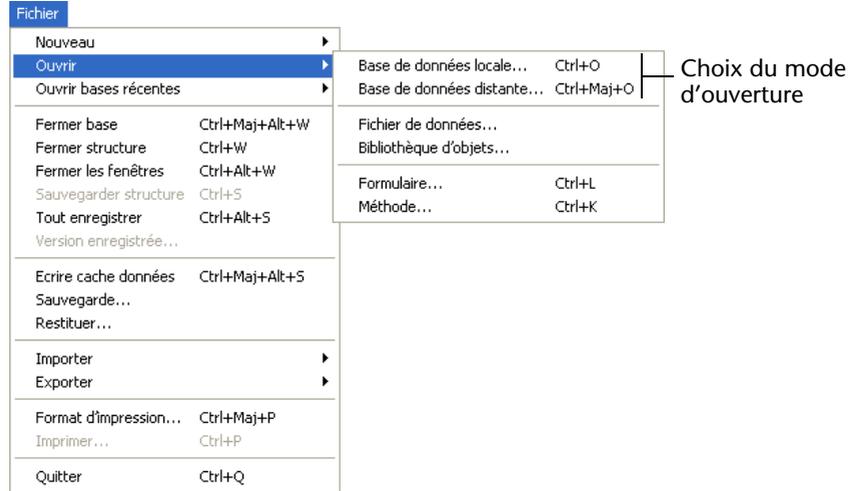
### Nouvelle architecture

En version 11, l'architecture client/serveur de 4D a été simplifiée, elle se traduit notamment par la fusion des applications 4D Client et 4D monoposte (*4D Developer*). Vous utilisez désormais une seule application, **4D**, pour vous connecter à des bases locales ou à des bases distantes exécutées par 4D Server. Dans le premier cas, l'application travaille en mode local (monoposte). Dans le second cas, elle travaille en mode distant (client/serveur).



L'utilisation de l'un ou l'autre des deux modes requiert une licence adéquate.

La sélection du mode d'utilisation s'effectue lors de la sélection de la base, par le choix du type de base à ouvrir :



- **Ouvrir/Base de données locale** : affiche la boîte de dialogue standard d'ouverture de documents, permettant de sélectionner un fichier de structure 4D.
- **Ouvrir/Base de données distante** : affiche la boîte de dialogue de connexion à 4D Server, permettant de sélectionner une base de données publiée.

*Note* Les fichiers .pth, permettant d'accéder directement à des bases 4D Server, ont été remplacés par des fichiers XML suffixés .4DLink. Pour plus d'informations, reportez-vous au [paragraphe "Enregistrement de l'accès à une base"](#), page 37.

## Gestion des dossiers distants

### Dossiers Resources et Extras

Le transfert de données personnalisées (images, fichiers, sous-dossiers...) entre le poste serveur et les postes clients doit désormais être effectué via le dossier **Resources**, situé à côté du fichier de structure de la base. Le contenu de ce dossier est mis à jour automatiquement sur chaque client au moment de sa connexion. Tous les mécanismes de référencement associé au dossier Resources sont pris en charge en mode client/serveur (dossier .lproj, XLIFF, images...).

En outre, 4D v11 SQL fournit divers outils permettant de gérer et de mettre à jour dynamiquement ce dossier, notamment un nouvel Explorateur de ressources (cf. [paragraphe “Explorateur de ressources”](#), page 18).

Par conséquent, à compter de la version 11.2 de 4D v11 SQL, l'utilisation du dossier **Extras** est déconseillée car il ne bénéficie pas des mêmes mécanismes que le dossier Resources. Désormais, vous devez utiliser le dossier Resources en lieu et place du dossier Extras. Ce dernier reste toutefois pris en charge par 4D Server afin de préserver la compatibilité des applications existantes.

---

*Note* L'emplacement des dossiers internes téléchargés sur les postes 4D distants (clients) a été modifié dans 4D v11 SQL. Pour plus d'informations, reportez-vous au [paragraphe “Dossier 4D”](#), page 45.

---

## Dossier Components

En mode client/serveur, le développement et l'installation de composants s'effectue comme en mode local, à une exception près : en client/serveur, il n'est pas possible d'utiliser des raccourcis (Windows) ou des alias (Mac OS) vers des fichiers de composants à l'intérieur du dossier **Components** d'une base.

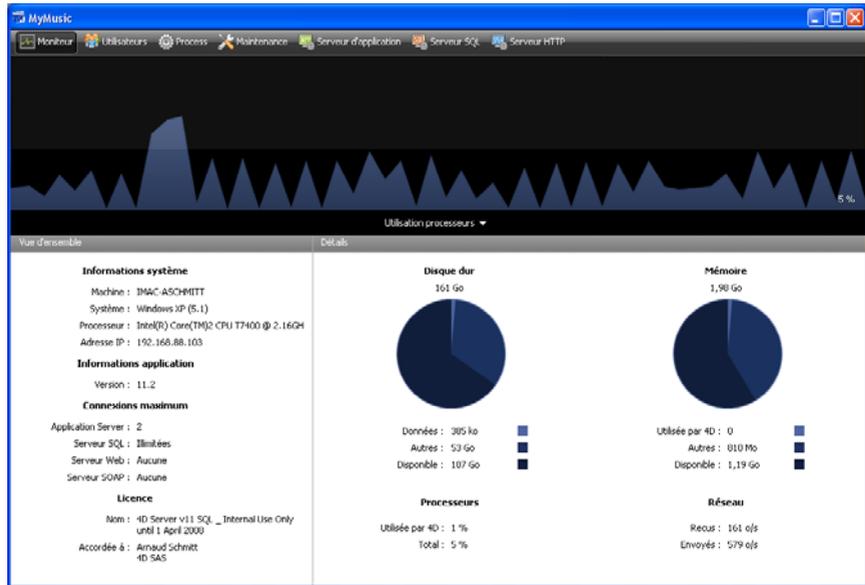
Côté client, le dossier Components est téléchargé au même emplacement que les autres éléments de la structure, c'est-à-dire :

Système	Emplacement dossier Components sur le poste client
Mac OS	{Disque}:/users/{nom_utilisateur}/library/Caches/ 4D:NomDeLaBase_Adresse:Components
Windows XP	{Disque}:\Documents and Settings\{nom_utilisateur}\ Local Settings\Application Data\NomDeLaBase_Adresse\ Components
Windows Vista	{Disque}:\Users\{nom_utilisateur}\AppData\Local\4D\ NomDeLaBase_Adresse\Components

La hiérarchie d'origine du dossier **Components** est reconstituée sur chaque poste client.

## Nouvelle fenêtre d'administration

4D Server v11 SQL bénéficie d'une nouvelle fenêtre d'administration plus complète et plus ergonomique :



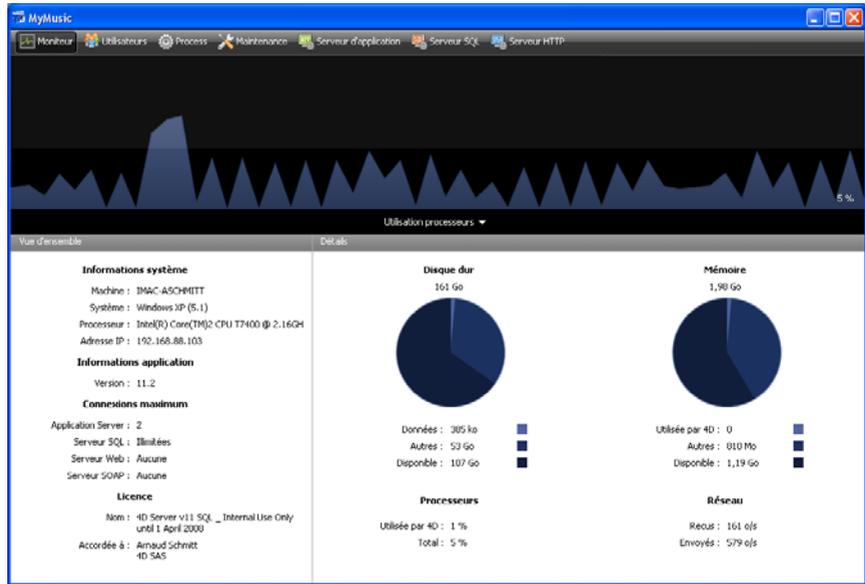
Cette fenêtre propose différents outils d'analyse et de contrôle de la base de données publiée. La fenêtre comporte sept pages, accessibles via une zone de boutons située dans la partie supérieure :



Chaque page est détaillée dans les paragraphes suivants.

## Moniteur

La page **Moniteur** affiche des informations dynamiques relatives à l'exploitation de la base de données ainsi que des informations sur le système et l'application 4D Server :



*Note* Sous Windows, l'affichage de ces informations est lié aux droits de l'utilisateur ayant ouvert la session. Pour plus d'informations, reportez-vous au [paragraphe "Affichage des informations du Moniteur \(Windows\)"](#), page 89.

## Zone graphique

La zone graphique permet de visualiser l'évolution en temps réel de trois types de paramètres : le taux d'utilisation des processeurs, le trafic réseau et l'occupation de la mémoire. Vous sélectionnez le paramètre à afficher via le menu situé au centre la fenêtre :

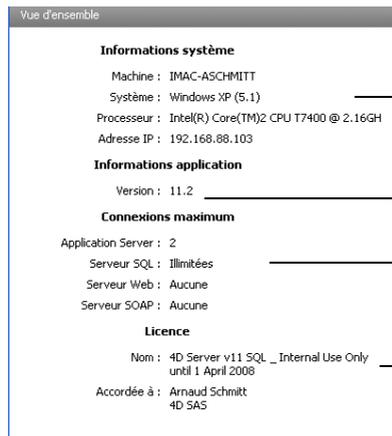


- **Utilisation processeurs** : Taux d'utilisation globale du ou des processeur(s) de la machine, toutes applications confondues. La part spécifique de 4D Server dans ce taux d'utilisation est fournie dans la zone d'informations "Processeurs".

- **Réseau** : Nombre d'octets reçus par seconde par 4D Server.  
Le nombre d'octets envoyés par 4D Server est fourni dans la zone d'informations "Réseau".
- **Mémoire** : Quantité de mémoire RAM de la machine utilisée par 4D Server.  
Une vue plus détaillée de l'utilisation de la mémoire est fournie dans la zone d'informations "Mémoire".

### Zone Vue d'ensemble

La zone "Vue d'ensemble" fournit diverses informations relatives au système, à l'application et aux licences installées sur la machine de 4D Server.



Ordinateur, système et adresse IP du serveur

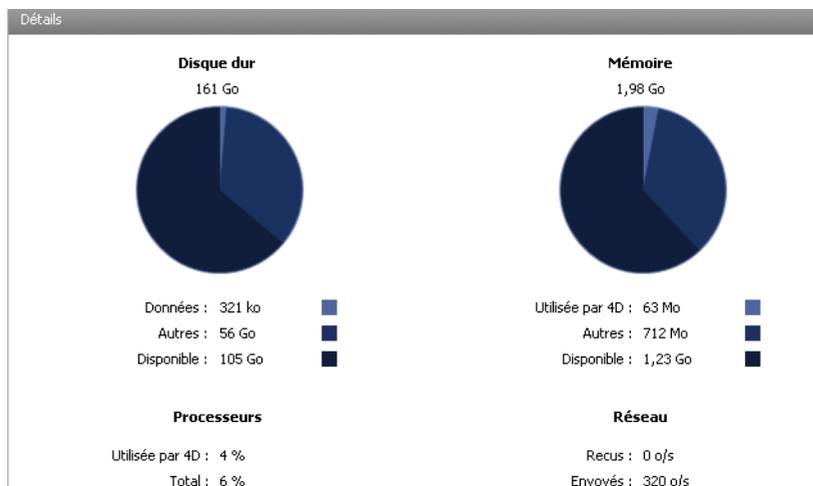
Numéro de version interne de 4D Server

Nombre de connexions simultanées autorisées par type de serveur

Description de la licence

### Zone Détails

La zone "Détails" reprend une partie des informations affichées dans la zone graphique et propose des informations complémentaires.



- **Disque dur** : Capacité globale du disque dur et répartition entre l'espace occupé par les données de la base (fichier de données + index des données), l'espace occupé par les autres fichiers et l'espace disponible.
- **Mémoire** : Mémoire RAM installée sur la machine et quantité de mémoire occupée par 4D Server, par les autres applications ainsi que mémoire disponible.  
La mémoire occupée par 4D Server peut également être affichée dynamiquement dans la zone graphique.
- **Processeurs** : Taux instantané d'occupation du ou des processeurs(s) de la machine par 4D Server et par les autres applications. Ce taux est recalculé en permanence.  
Le taux d'occupation par 4D Server peut également être affiché dynamiquement dans la zone graphique.
- **Réseau** : Nombre instantané d'octets reçus via le réseau par 4D Server nombre d'octets envoyés par l'application. Cette valeur est réactualisée en permanence.  
Le nombre d'octets reçus par 4D Server peut également être affiché dynamiquement dans la zone graphique.

### Affichage des informations du Moniteur (Windows)

Sous Windows, certaines informations système affichées dans la page **Moniteur** sont récupérées via les outils de l'"Analyseur de Performance" de Windows. L'accès à ces outils n'est permis que si l'utilisateur ayant ouvert la session à partir de laquelle a été lancée 4D Server dispose des autorisations nécessaires. Cet utilisateur doit soit :

- appartenir au groupe "Administrateurs",
- sous Windows Vista : appartenir au groupe "Utilisateurs de l'Analyseur de Performances" (pour un utilisateur non administrateur)
- ▶ Pour placer un utilisateur non-administrateur dans le groupe "Utilisateurs de l'Analyseur de Performances" sous Windows Vista :

---

*Note* Vous devez utiliser un compte Administrateur pour effectuer les manipulations.

---

- 1 **Allez dans le Panneau de Configuration et ouvrez le panneau "Comptes d'utilisateurs".**
- 2 **Cliquez sur l'onglet "Options avancées" puis sur le bouton "Avancé" de la partie "Gestion avancée des utilisateurs".**  
L'application "lusrmgr" s'exécute.

- 3 Double-cliquez sur le dossier "Groupes".
- 4 Double-cliquez sur le groupe "Utilisateurs de l'Analyseur de Performances".  
Une fenêtre nommée "Propriétés de Utilisateurs de l'Analyseur de Performances" apparaît.
- 5 Cliquez sur le bouton **Ajouter...** afin d'ajouter un utilisateur.
- 6 Dans la zone de texte intitulée "Entrez les noms des objets à sélectionner", saisissez les noms des utilisateurs à autoriser.
- 7 Cliquez sur **OK** (deux fois).
- 8 Fermez l'application "Iusrmgr" puis les "Comptes d'utilisateurs".

## Utilisateurs

La page **Utilisateurs** liste les utilisateurs connectés à la base :

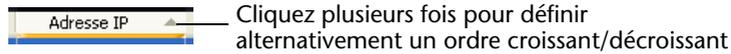
Utilisateur 4D	Nom de machine	Nom de session	Adresse IP	Connexion	Temps CPU	Activité
Super_Utilisateur	IMAC-ASCHMITT	Arnaud Schmitt	192.168.00.103	12/03/2008 11:34	00:00:01	0 %
Super_Utilisateur	IMAC-ESMITH	esmith	192.168.88.97	12/03/2008 11:36	00:00:01	0 %

Le bouton  indique entre parenthèses le nombre total d'utilisateurs connectés à la base (ce nombre ne tient pas compte des éventuels filtres d'affichage appliqués à la fenêtre).

La page contient également une zone de recherche dynamique et des boutons de commande.



Vous pouvez modifier l'ordre des colonnes par simple glisser-déposer de la zone d'en-tête des colonnes. Vous pouvez également trier la liste sur les valeurs d'une colonne en cliquant sur son en-tête :



## Liste des utilisateurs

Pour chaque utilisateur connecté à la base, la liste fournit les informations suivantes :

- *Système* de la machine cliente (Mac OS ou Windows) sous forme d'icône.
- *Utilisateur 4D* : Nom d'utilisateur 4D. Si les mots de passe ne sont pas activés, tous les utilisateurs sont nommés "Super\_Utilisateur".
- *Nom de machine* : Nom de l'ordinateur du poste client.
- *Nom de session* : Nom de la session ouverte sur le poste client.
- *Adresse IP* : Adresse IP de la machine cliente.
- *Connexion* : Date et heure de la connexion du poste client.
- *Temps CPU* : Temps processeur consommé par cet utilisateur depuis la connexion.
- *Activité* : Ratio du temps que 4D Server consacre à cet utilisateur (affichage dynamique).

## Zone de recherche/filtrage

Utilisateur;Machine;Session... 

Cette fonction permet de réduire le nombre de lignes affichées dans la liste à celles qui correspondent au texte saisi dans la zone de recherche. La zone indique les colonnes dans lesquelles la recherche/le filtrage sera effectué(e). Dans la page Utilisateurs, il s'agit des colonnes *Utilisateur 4D*, *Nom de machine* et *Nom de session*.

La mise à jour de la liste est effectuée en temps réel à mesure que vous saisissez du texte dans la zone.

Il est possible de saisir plus d'une valeur à chercher. Utilisez un point-virgule pour séparer les valeurs. L'opérateur utilisé dans ce cas est du type *OU*.

Par exemple, si vous saisissez "Jean;Marie;Pierre", seules les lignes comportant Jean OU Marie OU Pierre dans l'une des colonnes cibles seront conservées.

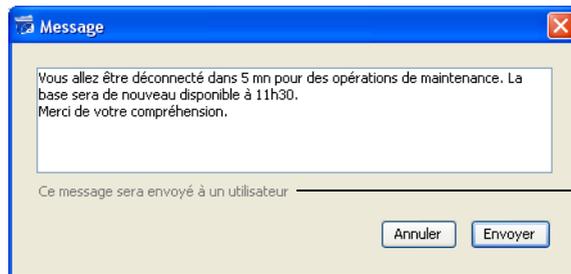
## Boutons d'administration

La page comporte trois boutons de commande. Ces boutons sont actifs si au moins une ligne est sélectionnée. Vous pouvez sélectionner plusieurs lignes en appuyant sur la touche **Maj** pour une sélection continue ou **Ctrl** (Windows) / **Commande** (Mac OS) pour une sélection discontinue.

### ■ Envoyer message

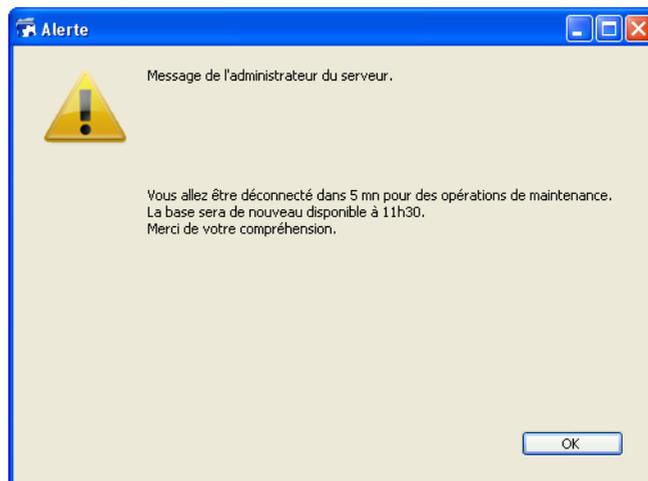
Ce bouton permet d'envoyer un message aux utilisateurs 4D sélectionnés dans la fenêtre. Si aucun utilisateur n'est sélectionné, le bouton est désactivé.

Lorsque vous cliquez sur le bouton, une boîte de dialogue apparaît, vous permettant saisir le message. La boîte de dialogue indique le nombre d'utilisateurs qui recevront le message ;



Nombre d'utilisateurs sélectionnés

Le message sera affiché sous forme d'alerte sur les postes clients :



### ■ Visualiser process

Ce bouton permet de visualiser directement les process du ou des utilisateur(s) sélectionné(s) dans la page **Process** de la fenêtre d'administration. Lorsque vous cliquez sur ce bouton, 4D Server bascule sur la page Process et pré-remplit la zone de recherche/filtrage

de cette page avec les noms des utilisateurs sélectionnés. Pour plus d'informations, reportez-vous à la description de cette page.

### ■ Déconnecter

Ce bouton permet de forcer la déconnexion du ou des utilisateur(s) sélectionné(s).

Lorsque vous cliquez sur ce bouton, une boîte de dialogue d'alerte apparaît, vous permettant de confirmer ou d'annuler l'opération.

*Note* Vous pouvez déconnecter directement les utilisateurs sélectionnés sans afficher la boîte de dialogue de confirmation : pour cela, appuyez sur la touche **Alt** (Windows) ou **Option** (Mac OS) avant de cliquer sur le bouton **Déconnecter**.

## Process

La page Process liste les process en cours d'exécution :

Type	Session	Nom de process	Num	État	Temps CPU	Activité
Serveur SQL		SHARED pool worker 1	0	En cours d'exécution	00:00:00	0,00 %
Serveur SQL		SQL Net Session Manager Thread	0	En cours d'exécution	00:00:00	0,00 %
Serveur d'application		Gestionnaire de client	3	En attente de sémaphore interne	00:00:05	0,00 %
Serveur d'application		Interface utilisateur	1	En attente d'entrée-sortie	00:03:05	34,29 %
Serveur d'application		Process minuteur interne	2	En cours d'exécution	00:05:58	0,00 %
Serveur d'application		Process passerelle interne	4	En attente de sémaphore interne	00:01:31	0,00 %
Process 4D Client	arnaud schmitt	arnaud schmitt	6	En attente d'entrée-sortie	00:00:02	0,00 %
Process base 4D Client	arnaud schmitt	arnaud schmitt	0	En cours d'exécution	00:00:00	0,00 %
Process base 4D Client	arnaud schmitt	P_11	0	En cours d'exécution	00:00:00	0,00 %
Process 4D Client	arnaud schmitt	P_11	0	En attente d'entrée-sortie	00:00:00	0,00 %
Process 4D Client	arnaud schmitt	Process développement	7	En attente d'entrée-sortie	00:00:02	0,00 %
Process base 4D Client	arnaud schmitt	Process développement	0	En cours d'exécution	00:00:00	0,00 %
Process base 4D Client	arnaud schmitt	Process principal	0	En cours d'exécution	00:00:00	0,00 %
Process 4D Client	arnaud schmitt	Process principal	5	En attente d'entrée-sortie	00:00:03	0,00 %

Le bouton  indique entre parenthèses le nombre total de process en cours d'exécution à la base (ce nombre ne tient pas compte des éventuels filtres d'affichage appliqués à la fenêtre ni de l'état de l'option **Afficher les process par groupes**).

Vous pouvez modifier l'ordre des colonnes par simple glisser-déposer de la zone d'en-tête des colonnes. Vous pouvez également trier la liste sur les valeurs d'une colonne en cliquant sur son en-tête.

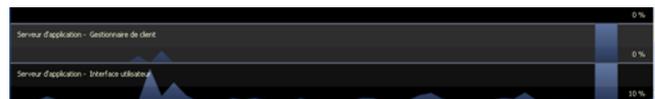
- Comme la page Utilisateurs, cette page contient une zone de recherche/filtrage dynamique, permettant de réduire le nombre de lignes affichées dans la liste à celles qui correspondent au texte saisi dans la zone de recherche. La recherche/le filtrage est effectué(e) dans les colonnes *Session* et *Nom de process*.



- Vous disposez également de trois boutons-raccourcis permettant de filtrer par famille les process affichés dans la fenêtre :



- **Process utilisateurs** : Process générés par et pour les sessions utilisateurs. Ces process sont précédés d'une icône en forme de personnage 🧑.
  - **Process 4D** : Process générés par le moteur de 4D Server. Ces process sont précédés d'une icône en forme de roue crantée ⚙️.
  - **Process en attente** : Process inactifs mais conservés temporairement et pouvant être réutilisés à tout moment. Ce mécanisme permet d'optimiser la réactivité de 4D Server. Ces process sont précédés d'une icône grisée en forme de personnage 🧑.
- L'option **Afficher les process par groupes** vous permet de regrouper les process internes de 4D Server ainsi que les process clients, pour plus de lisibilité. Lorsque vous cochez cette option :
    - les process clients 4D "jumeaux" (Process client 4D principal et Process base 4D client, cf. [paragraphe "Type du process", page 95](#)) sont groupés en un seul,
    - le groupe "Gestionnaires de tâches" est créé ; il inclut les process internes dédiés à la répartition des tâches (*Shared balancer*, *Net session manager*, *Exclusive pool worker*),
    - le groupe "Gestionnaires clients" est créé ; il inclut les différents process internes clients.
  - La zone inférieure de la fenêtre permet d'afficher la représentation graphique de l'activité du ou des process sélectionné(s) :



*Note* Vous pouvez sélectionner plusieurs lignes en appuyant sur la touche **Maj** pour une sélection continue ou **Ctrl** (Windows) / **Commande** (Mac OS) pour une sélection discontinue.

---

L'activité du process est le pourcentage du temps que 4D Server a consacré à ce process (ratio).

Comme dans les versions précédentes de 4D Server, la fenêtre fournit les informations suivantes pour chaque process :

- Type du process (cf. ci-dessous),
- Session (vide dans le cas d'un process 4D et nom de l'utilisateur 4D dans le cas d'un process utilisateur)
- Nom du process,
- Numéro du process (tel que retourné par la fonction Nouveau process par exemple),

---

*Note* Le numéro du process est le numéro attribué sur le serveur. Dans le cas d'un process global, ce numéro peut être différent de celui attribué sur le poste client.

---

- Etat courant du process,
- Temps (en secondes) d'exécution du process depuis sa création,
- Pourcentage du temps que 4D Server a consacré à ce process (ratio).

## Type du process

Chaque process est identifié par une icône et un type. La couleur et la forme de l'icône indique la famille du process :

### Icône Type de process (description)

	Serveur d'application
	Serveur SQL
	Serveur DB4D (moteur de base de données)
	Serveur Web
	Serveur SOAP
	Process client 4D protégé (process développement d'un 4D connecté)
	Process client 4D principal (process principal d'un 4D connecté. Process collaboratif, équivalent sur le serveur du process créé sur le poste client)
	Process base 4D client (process parallèle à un process 4D client. Process préemptif chargé de contrôler le process client 4D principal correspondant)
	Process en attente (ancien ou futur "Process client 4D base de données")

**Icône Type de process (description)**



Process 4D client (process tournant sur le 4D connecté)



Procédure stockée (process lancé par un 4D connecté et tournant sur le serveur)



Méthode Web (lancée par un 4DACTION par exemple)



Méthode SOAP (lancée par un Web Service)



Méthode SQL (lancée par une requête SQL)

---

*Note* Chaque process client 4D principal et son process base 4D client "jumeau" sont regroupés lorsque l'option **Afficher les process par groupes** est cochée.

---

**Boutons d'administration**

La page comporte cinq boutons de commande permettant d'agir sur le ou les process sélectionné(s). A noter que vous ne pouvez agir que sur les process utilisateurs.



Tuer process



Endormir process



Réactiver process



Tracer Process



Voir utilisateurs

- **Tuer process** : permet de tuer le ou les process sélectionné(s). Lorsque vous cliquez sur ce bouton, une boîte de dialogue d'alerte apparaît, vous permettant de confirmer ou d'annuler l'opération.
- 

*Note* Vous pouvez tuer directement les process sélectionnés sans afficher la boîte de dialogue de confirmation : pour cela, appuyez sur la touche **Alt** (Windows) ou **Option** (Mac OS) avant de cliquer sur le bouton.

---

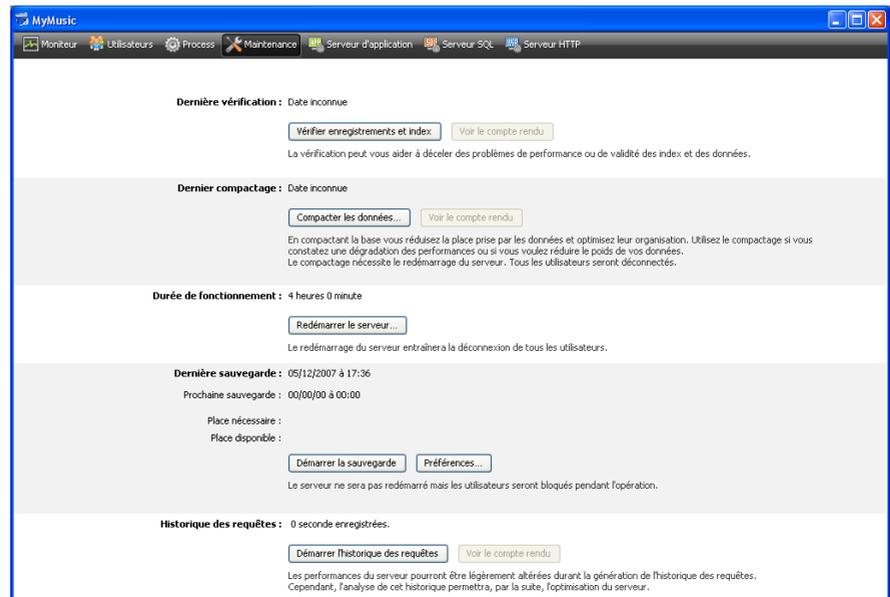
- **Endormir process** : permet d'endormir le ou les process sélectionné(s).
- **Réactiver process** : permet de réactiver le ou les process sélectionné(s). Les process doivent avoir été auparavant endormis (via le bouton précédent ou par programmation) sinon le bouton est sans effet.
- **Tracer process** : permet d'ouvrir sur le poste serveur une ou plusieurs fenêtre(s) du débogueur pour le ou les process sélectionné(s). Lorsque vous cliquez sur ce bouton, une boîte de dialogue d'alerte apparaît, vous permettant de confirmer ou d'annuler l'opération.  
A noter que la fenêtre du débogueur ne s'affiche que lorsque du code 4D est effectivement exécuté sur le poste serveur (par exemple dans le cadre d'un trigger ou de l'exécution d'une méthode ayant l'attribut "Exécuter sur serveur").

*Note* Vous pouvez déboguer un process directement, sans afficher la boîte de dialogue de confirmation : pour cela, appuyez sur la touche **Alt** (Windows) ou **Option** (Mac OS) avant de cliquer sur le bouton.

- **Voir utilisateurs** : permet d'afficher dans la page Utilisateurs tous les process du ou des utilisateur(s) sélectionné(s). Le bouton est actif lorsqu'un process utilisateur au moins est sélectionné.

## Maintenance

La page Maintenance fournit diverses informations relatives au fonctionnement courant de la base. Elle donne également accès aux fonctions de maintenance élémentaires :



- **Dernière vérification** : Cette zone indique la date, l'heure et le statut de la dernière vérification des données effectuée sur la base. Pour plus d'informations sur la procédure de vérification des données, reportez-vous au manuel *Mode Développement*.

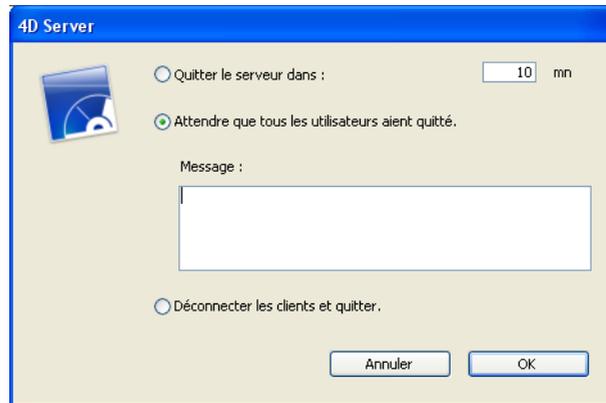
Le bouton **Vérifier enregistrements et index** permet de lancer directement l'opération de vérification, sans interruption du serveur. A noter que le serveur peut être sensiblement ralenti durant l'opération. Tous les enregistrements et tous les index de la base sont vérifiés. Si vous souhaitez pouvoir cibler la vérification ou disposer d'options supplémentaires, vous devez utiliser le Centre de sécurité et de maintenance (CSM).

A l'issue de la vérification, un fichier de compte-rendu est généré aux

formats xml et html sur le serveur, dans le dossier **Logs** placé à côté du fichier de structure de la base. Le bouton **Voir le compte-rendu** (nommé **Télécharger le compte-rendu** si l'opération a été effectuée depuis un client) vous permet d'afficher le fichier dans votre navigateur.

- **Dernier compactage** : Cette zone indique la date, l'heure et le statut du dernier compactage des données de la base. Pour plus d'informations sur la procédure de compactage des données, reportez-vous au manuel *Mode Développement*.

Le bouton **Compacter les données** permet de lancer directement une opération de compactage des données. Cette opération nécessite de stopper le serveur : lorsque vous cliquez sur le bouton, la boîte de dialogue de fermeture de la base 4D Server apparaît, vous permettant de choisir le mode d'interruption de l'exploitation :



Après l'interruption effective de la base, 4D Server effectue un compactage standard des données de la base. Si vous souhaitez disposer d'options supplémentaires, vous devez utiliser le Centre de sécurité et de maintenance (CSM).

Une fois le compactage terminé, 4D Server relance automatiquement la base. Les utilisateurs 4D peuvent alors se reconnecter.

---

*Note* Si la demande de compactage a été effectuée depuis un client 4D distant, ce poste est automatiquement reconnecté par 4D Server.

---

Un fichier de compte-rendu est généré aux formats xml et html sur le serveur, dans le dossier **Logs** placé à côté du fichier de structure de la base. Le bouton **Voir le compte-rendu** (nommé **Télécharger le compte-rendu** si l'opération a été effectuée depuis un client) vous permet d'afficher le fichier dans votre navigateur.

- **Durée de fonctionnement** : Cette zone indique la durée de fonctionnement du serveur depuis son dernier démarrage (jours, heures et minutes).  
Le bouton **Redémarrer le serveur** vous permet de provoquer un redémarrage immédiat du serveur. Lorsque vous cliquez sur ce bouton, la boîte de dialogue de fermeture de la base 4D Server apparaît, vous permettant de choisir le mode d'interruption de l'exploitation. Après le redémarrage, 4D Server relance automatiquement la base. Les utilisateurs 4D peuvent alors se reconnecter.

---

*Note* Si la demande de redémarrage a été effectuée depuis un client 4D distant, ce poste est automatiquement reconnecté par 4D Server.

---

- **Dernière sauvegarde automatique** : Cette zone indique la date et l'heure de la dernière sauvegarde automatique de la base et fournit des informations relatives à la prochaine sauvegarde. Les sauvegardes automatiques sont paramétrées via la page "Périodicité" des préférences de la base.
  - *Prochaine sauvegarde* : date et heure de la prochaine sauvegarde automatique.
  - *Place nécessaire estimée* : estimation de la taille nécessaire pour la sauvegarde. La taille réelle du fichier de sauvegarde pourra varier en fonction des paramétrages (compression...) et des variations du fichier de données.
  - *Place disponible* : place disponible sur le volume de sauvegarde.

Le bouton **Sauvegarder la base** permet de démarrer une sauvegarde immédiate de la base en utilisant les paramètres de sauvegarde courants (fichiers sauvegardés, emplacement des archives, options, etc.). Vous pouvez visualiser ces paramètres en cliquant sur le bouton **Préférences...** Pendant une sauvegarde sur le serveur, les postes clients sont "bloqués" (mais pas déconnectés) et il n'est pas possible à de nouveaux clients de se connecter.

- **Historique des requêtes** : Cette zone indique la durée d'enregistrement de l'historique des requêtes, lorsqu'il est activé. Le fichier d'historique des requêtes stocke diverses informations concernant les requêtes reçues par le serveur (hors requêtes Web) : heure, numéro de process, utilisateur, taille de la requête, durée de traitement, etc., permettant d'analyser le fonctionnement du serveur. Ce fichier est nommé *4DRequestsLog\_N* (N étant le numéro séquentiel du fichier) et est stocké dans le dossier **Logs** de la base. Une fois que le fichier atteint une taille

de 10 Mo, il est refermé et un nouveau fichier est généré, avec un numéro séquentiel incrémenté.

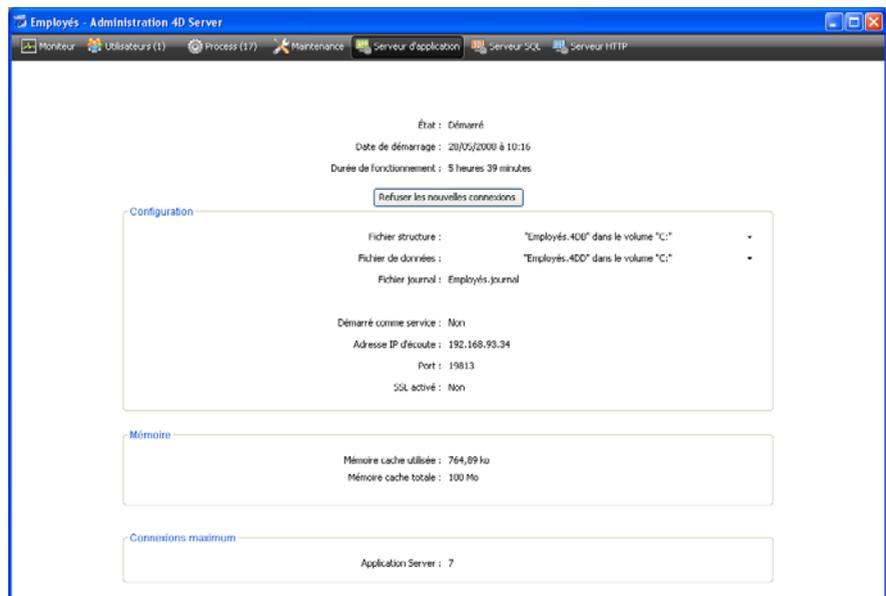
Le bouton **Démarrer l'historique des requêtes** permet de générer un nouveau fichier et d'activer le mode d'enregistrement des requêtes. Ce mode pouvant dégrader sensiblement les performances du serveur, il est à réserver à la phase de mise au point de l'application. Une fois l'enregistrement des requêtes activé, le libellé du bouton devient **Arrêter l'historique des requêtes**, permettant de stopper l'enregistrement des requêtes à tout moment. A noter qu'une reprise de l'enregistrement des requêtes après un arrêt "écrase" le fichier précédent.

*Note* Il est possible de démarrer et de stopper l'historique des requêtes par programmation via la commande `FIXER PARAMETRE BASE`.

Le bouton **Voir le compte-rendu** (nommé **Télécharger le compte-rendu** si l'opération a été effectuée depuis un client) permet d'ouvrir une fenêtre système affichant le fichier d'historique des requêtes.

## Serveur d'application

La page Serveur d'application regroupe les informations relatives à la base de données publiée par 4D Server et permet de gérer cette publication :



- Informations de statut** La partie supérieure de la page fournit des informations sur le statut courant du serveur d'application de 4D Server.
- **Etat** : Démarré ou Arrêté
  - **Date de démarrage** : Date et heure de lancement de la base serveur. Cette date correspond à l'ouverture de la base par 4D Server.
  - **Durée de fonctionnement** : Durée écoulée depuis la dernière ouverture de la base.

**Bouton Refuser /  
Accepter nouvelles  
connexions**

Ce bouton fonctionne en bascule. Il permet de gérer l'accès de nouveaux postes clients à l'application serveur.

- Par défaut, lorsque la base est publiée :
  - Le libellé du bouton est "Refuser nouvelles connexions".
  - De nouveaux clients peuvent se connecter librement (dans les limites des connexions accordées par la licence).
  - Le nom de la base est publié dans la boîte de dialogue de connexion (si l'option "Publier le nom de la base au démarrage dans le dialogue de connexion" est cochée dans les Préférences).
- Si vous cliquez sur le bouton **Refuser nouvelles connexions** :
  - Le libellé du bouton devient "Accepter nouvelles connexions".
  - Plus aucun *nouveau* client ne peut alors se connecter.
  - Le nom de la base n'apparaît plus dans la boîte de dialogue de connexion.
  - Les clients déjà connectés ne sont pas déconnectés et peuvent continuer à travailler normalement.
  - Si vous cliquez sur le bouton **Accepter nouvelles connexions**, la base retourne dans l'état "par défaut".

Cette fonction permet par exemple à un administrateur, juste après avoir démarré le serveur, d'effectuer diverses opérations de maintenance (vérification, compactage...). S'il utilise une connexion cliente, il a la certitude d'être le seul à modifier les données. Il est également possible d'utiliser cette fonction en préparation d'une opération de maintenance nécessitant qu'aucun poste client ne soit connecté.

### Configuration

Cette zone fournit plusieurs informations sur la base de données 4D publiée par le serveur : nom et emplacement des fichiers de structure et de données et nom du fichier d'historique de la base. Vous pouvez cliquer sur le nom du fichier de structure ou de données afin de visualiser son chemin d'accès complet :



La partie inférieure de la zone indique les paramètres de configuration du serveur (démarré comme service, port et adresse IP) et l'activation du SSL pour les connexions client-serveur (ne concerne pas les connexions SQL ni Web).

### Mémoire

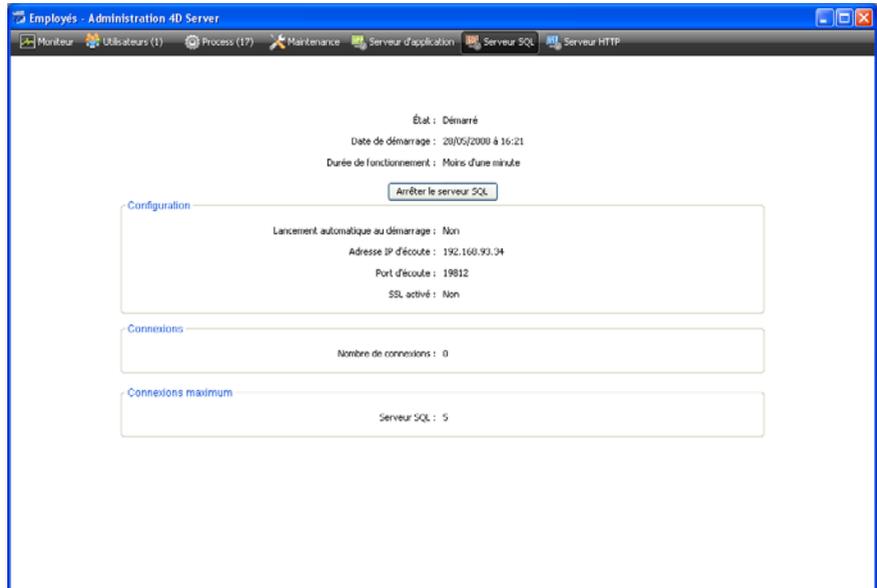
Cette zone indique la **mémoire cache totale** (paramètre défini dans les préférences de la base) et la **mémoire cache utilisée** (allocation dynamique par 4D Server en fonction des besoins).

### Connexions maximum

Nombre maximum de connexions clientes simultanées autorisées. Cette valeur dépend de la licence installée sur le poste serveur.

## Serveur SQL

La page Serveur SQL regroupe les informations relatives au serveur SQL intégré de 4D Server. Elle comporte également un bouton permettant de contrôler l'activation du serveur :



**Informations de statut** La partie supérieure de la page fournit des informations sur le statut courant du serveur SQL de 4D Server.

- **Etat** : Démarré ou Arrêté
- **Date de démarrage** : Date et heure du dernier lancement du serveur SQL. Cette valeur peut différer de celle du serveur d'application, si le lancement du serveur SQL ne s'effectue pas "au démarrage".
- **Durée de fonctionnement** : Délai écoulé depuis le dernier démarrage du serveur SQL.

### Bouton Démarrer / Arrêter le serveur SQL

Ce bouton fonctionne en bascule. Il permet de contrôler l'activation du serveur SQL de 4D Server.

- Lorsque l'état du serveur SQL est "Démarré", le bouton est libellé **Arrêter le serveur SQL**. Si vous cliquez sur ce bouton, le serveur SQL de 4D Server est immédiatement stoppé, il ne répond plus aux requêtes SQL externes reçues sur le port TCP désigné.

- Lorsque l'état du serveur SQL est "Arrêté", le bouton est libellé **Démarrer le serveur SQL**. Si vous cliquez sur ce bouton, le serveur SQL de 4D Server est immédiatement démarré, il répond aux requêtes SQL externes reçues sur le port TCP désigné. A noter que vous devez disposer d'une licence adéquate pour pouvoir exploiter le serveur SQL de 4D.

---

*Note* Le serveur SQL peut également être lancé automatiquement au démarrage de l'application (option des Préférences) ou par programmation.

---

### Configuration

Cette zone fournit plusieurs informations sur les paramètres de configuration du serveur SQL : lancement automatique au démarrage, adresse IP d'écoute, port TCP (19812 par défaut) et activation du SSL pour les connexions SQL (ne concerne pas les connexions 4D ni Web).

Ces paramètres peuvent être modifiés via les Préférences de 4D.

### Connexions

Nombre de connexions SQL actuellement ouvertes sur 4D Server.

### Connexions maximum

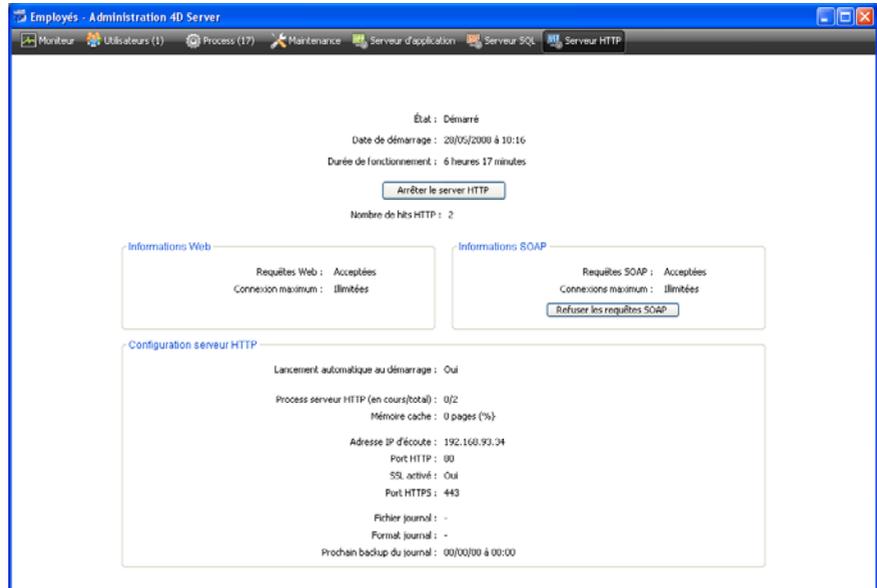
Nombre maximum de connexions SQL simultanées autorisées. Cette valeur dépend de la licence installée sur le poste serveur.

### Serveur HTTP

La page Serveur HTTP regroupe les informations relatives au fonctionnement du serveur Web et du serveur SOAP de 4D Server.

Le serveur Web permet de publier du contenu Web tel que des pages HTML ou des images à destination de navigateurs Web. Le serveur SOAP gère la publication de Web Services. Ces deux serveurs s'appuient sur le serveur HTTP interne de 4D Server.

La page comporte également des boutons permettant de contrôler l'activation des serveurs :



**Informations de statut** La partie supérieure de la page fournit des informations sur le statut courant du serveur HTTP de 4D Server.

- **Etat** : Démarré ou Arrêté
- **Date de démarrage** : Date et heure du dernier lancement du serveur HTTP. Cette valeur peut différer de celle du serveur d'application, si le lancement du serveur HTTP ne s'effectue pas "au démarrage".
- **Durée de fonctionnement** : Délai écoulé depuis le dernier démarrage du serveur HTTP.
- **Nombre de hits HTTP** : nombre de hits HTTP (bas niveau) reçus par le serveur HTTP depuis son démarrage.

**Bouton Démarrer / Arrêter le serveur HTTP** Ce bouton fonctionne en bascule. Il permet de contrôler l'activation du serveur HTTP de 4D Server.

- Lorsque l'état du serveur HTTP est "Démarré", le bouton est libellé **Arrêter le serveur HTTP**. Si vous cliquez sur ce bouton, le serveur HTTP de 4D Server est immédiatement stoppé, le serveur Web et le serveur SOAP n'acceptent plus aucune requête.

- Lorsque l'état du serveur HTTP est "Arrêté", le bouton est libellé **Démarrer le serveur HTTP**. Si vous cliquez sur ce bouton, le serveur HTTP de 4D Server est immédiatement démarré : les requêtes Web et les requêtes SOAP sont acceptées (à noter qu'il est possible de stopper séparément le serveur SOAP, cf. [paragraphe "Informations SOAP", page 106](#)).

- Notes*
- Vous devez disposer d'une licence appropriée pour pouvoir démarrer le serveur HTTP.
  - Le serveur HTTP peut également être lancé automatiquement au démarrage de l'application (option des Préférences) ou par programmation.
- 

### Informations Web

Cette zone fournit des informations spécifiques relatives au serveur Web de 4D Server.

- **Requêtes Web** : Acceptées ou Refusées. Cette information indique si le serveur Web est actif. Le serveur Web étant directement relié au serveur HTTP, les requêtes Web sont acceptées lorsque le serveur HTTP est démarré et refusées lorsqu'il est stoppé.
- **Connexions maximum** : Nombre maximum de connexions Web autorisées. Cette valeur dépend de la licence installée sur le poste serveur.

### Informations SOAP

Cette zone fournit des informations spécifiques relatives au serveur SOAP de 4D Server et contient un bouton de contrôle.

- **Requêtes SOAP** : Acceptées ou Refusées. Cette information indique si le serveur SOAP est actif. Pour que les requêtes SOAP soient acceptées, le serveur HTTP doit être démarré et le serveur SOAP doit explicitement accepter les requêtes (cf. bouton Accepter/Refuser).
- **Connexions maximum** : Nombre maximum de connexions SOAP autorisées. Cette valeur dépend de la licence installée sur le poste serveur.
- Bouton **Accepter/Refuser les requêtes SOAP** : Ce bouton fonctionne en bascule. Il permet de contrôler l'activation du serveur SOAP de 4D Server. Ce bouton modifie la valeur de l'option **Autoriser requêtes SOAP** dans la page "Web Services/SOAP" des préférences (et inversement).  
Si vous cliquez sur le bouton **Accepter les requêtes SOAP** et que le serveur HTTP est arrêté, 4D le démarre automatiquement.

**Configuration serveur HTTP**

Cette zone fournit plusieurs informations sur les paramètres de configuration et le fonctionnement du serveur HTTP :

- **Lancement automatique au démarrage** : paramètre défini via les Préférences de 4D Server.
- **Process serveur HTTP (en cours/total)** : nombre de process HTTP créés sur le serveur (nombre courant de process / cumul de tous les process créés)
- **Mémoire cache (utilisée/totale)** : taille de la mémoire cache du serveur HTTP, lorsqu'elle est activée (taille réellement occupée par le cache / taille maximale théorique allouée au cache dans les Préférences).
- **Adresse IP d'écoute, Port TCP (80 par défaut), SSL activé** pour les connexions HTTP (ne concerne pas les connexions 4D ni SQL) et **Port HTTPS utilisé** : paramètres de configuration courants du serveur HTTP, définis dans la page **Web/Configuration** des Préférences.
- **Informations sur le fichier journal** : emplacement, format et date de la prochaine sauvegarde automatique du journal du serveur HTTP (fichier *logweb.txt*).

## Nouvelles fonctions accessibles à distance

Avec 4D v11 SQL, il est désormais possible d'administrer le poste 4D Server et de compiler une base depuis un 4D distant (poste client).

### Administration de 4D Server depuis un poste distant

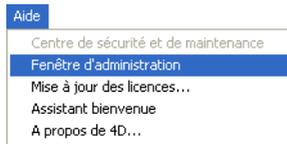
Avec 4D v11 SQL, vous pouvez administrer le poste 4D Server depuis un 4D distant (poste client). Le principe consiste à ouvrir la fenêtre d'administration de 4D Server (cf. [paragraphe "Nouvelle fenêtre d'administration"](#), page 86) sur le poste client.

### Ouvrir la fenêtre d'administration sur un poste client

Pour ouvrir une fenêtre d'administration du serveur depuis un poste client, vous devez être connecté à la base distante en tant que **Super\_Utilisateur** ou **Administrateur**. Dans le cas contraire, si vous tentez d'ouvrir la fenêtre d'administration, une erreur de privilège (-9991) est générée.

Vous disposez des possibilités suivantes :

- Choisir la commande **Fenêtre d'administration** dans le menu **Aide** ou cliquer sur le bouton correspondant dans la barre d'outils de 4D :

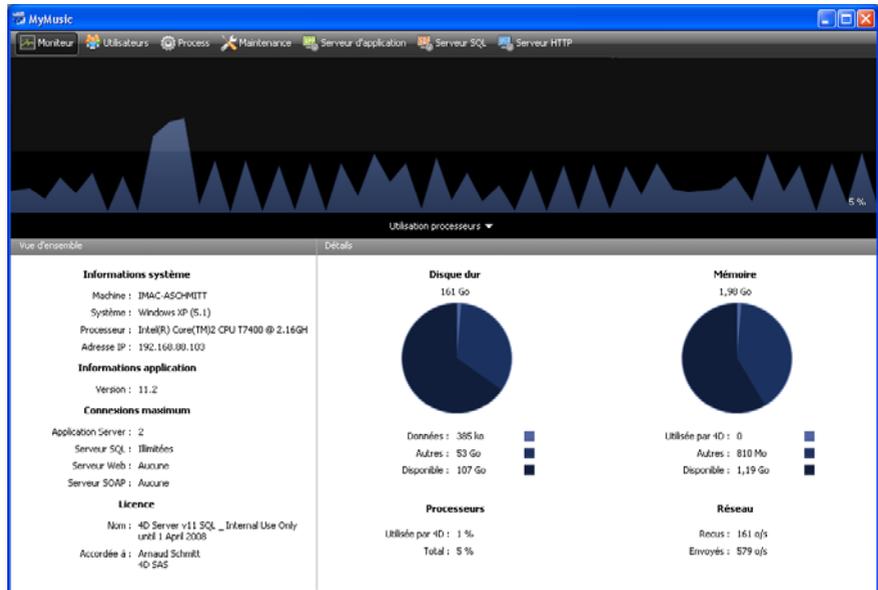


Nouveau bouton de la barre d'outils de 4D

Nouvelle commande du menu Aide de 4D

- Exécuter la nouvelle commande OUVRIR FENETRE ADMINISTRATION (cf. [commande OUVRIR FENETRE ADMINISTRATION, page 43](#)).

Une fenêtre d'administration du serveur s'affiche alors sur le poste client :



### Spécificités de l'administration via un poste client

Un poste client affichant la fenêtre d'administration du serveur accède à toutes les informations disponibles et peut agir sur les process et le démarrage des serveurs.

Il existe cependant des restrictions et des particularités de fonctionnement lorsque la fenêtre d'administration du serveur est affichée sur un poste client :

- Dans la page [Process](#), il n'est pas possible de tracer un process utilisateur (la fenêtre du débogueur apparaîtrait sur le poste serveur).
- Dans la page [Maintenance](#), il est possible d'exécuter des actions provoquant la déconnexion de tous les clients et le redémarrage du serveur (compactage et redémarrage). Dans ce cas, le poste client ayant demandé l'opération est automatiquement reconnecté à l'issue du redémarrage.
- Dans la page [Maintenance](#), il est possible d'accéder aux préférences de sauvegarde, toutefois seuls les paramètres des pages Périodicité, Sauvegarde et Restitution sont modifiables.
- Dans la page [Maintenance](#), les boutons **Voir le compte rendu** sont renommés **Télécharger le compte rendu** après exécution d'une opération de maintenance. Ces fichiers sont téléchargés dans le dossier de la base sur le poste client avant d'être affichés.

## Compilation depuis les postes distants

Il est désormais possible de compiler une application 4D à partir d'une connexion distante. Autrement dit, il est possible de compiler depuis un client 4D. Dans les versions précédentes, la compilation pouvait uniquement être effectuée depuis l'application monoposte 4D Developer.

*Note* En revanche, il n'est pas possible de construire une application 4D personnalisée (monoposte ou client/serveur) à partir d'une connexion distante. Le Générateur d'applications n'est pas accessible dans cet environnement (la ligne de menu est grisée).

Côté client, l'interface et les principes de compilation sont identiques à ceux des versions monopostes. La compilation peut être déclenchée soit depuis le menu **Développement** ou la barre d'outils, soit depuis la fenêtre du compilateur :



*Note* La licence "4D Team Server" est nécessaire côté 4D Server pour que les postes clients puissent accéder à la fonction de compilation.

---

Un seul poste client peut compiler la base à un instant donné. La compilation par un client verrouille la fonction pour les autres postes. Si un autre poste client tente de compiler la base au même moment, un message d'alerte apparaît.

Pendant qu'un poste client effectue une compilation, les autres clients peuvent continuer de travailler et de modifier des méthodes ou tout autre élément de structure. Le code compilé et le code interprété seront alors différents, ce qui nécessitera de recompiler la base ultérieurement.

Le code compilé est envoyé dans le fichier .4DB sur le serveur au fur et à mesure de la compilation.

Côté client, après la fin de l'opération, il est possible de redémarrer le serveur en mode interprété ou en mode compilé en utilisant les commandes correspondantes dans le menu **Exécution**. Lorsqu'un client demande le redémarrage du serveur en compilé/en interprété, la boîte de dialogue standard d'arrêt du serveur apparaît, permettant d'accorder un délai ou d'envoyer un message d'avertissement aux autres clients. Lorsque le serveur a redémarré, le client à l'origine du redémarrage est automatiquement reconnecté.

Côté serveur, le redémarrage en compilé/interprété nécessite d'utiliser la boîte de dialogue standard d'ouverture de fichier (pop up menu associé au bouton **Ouvrir**).

*Note* La compilation sur réseau WAN est déconseillée pour des raisons de performances (notamment dans le cas de bases comportant de nombreuses méthodes) car l'opération génère une quantité importante d'échanges réseau.

---

### Accès aux préférences de sauvegarde depuis les postes distants

Il est désormais possible de modifier la plupart des préférences de sauvegarde de la base depuis un poste distant. Tous les paramètres des pages Périodicité, Sauvegarde et Restitution sont modifiables depuis un poste distant disposant des privilèges d'accès suffisants.

Seules les préférences liées aux fichiers sauvegardés et à leur emplacement sur disque (page Configuration) restent modifiables sur le poste serveur uniquement.

## 4D Server v11 SQL et le langage

Outre la nouvelle propriété de méthode (cf. [paragraphe “Nouvel attribut de méthode : Exécuter sur serveur”](#), page 14), 4D Server v11 SQL inclut des optimisations relatives à l’utilisation des commandes du langage.

### Exécution des commandes par formule sur le serveur

Dans une base de données créée avec 4D v11, pour des raisons d’optimisation les commandes de recherches et de tris "par formule" sont dorénavant exécutées sur le serveur. Seul le résultat est retourné au client. Il s’agit des commandes suivantes :

CHERCHER PAR FORMULE

CHERCHER PAR FORMULE DANS SELECTION

TRIER PAR FORMULE

En cas d’appel direct de variables dans la formule, la requête est calculée avec la valeur de la variable sur le poste client. Par exemple, `CHERCHER PAR FORMULE([latable];[latable]lechamp=lavariab)` sera exécutée sur le serveur mais avec le contenu de la variable *mavariab* du client.

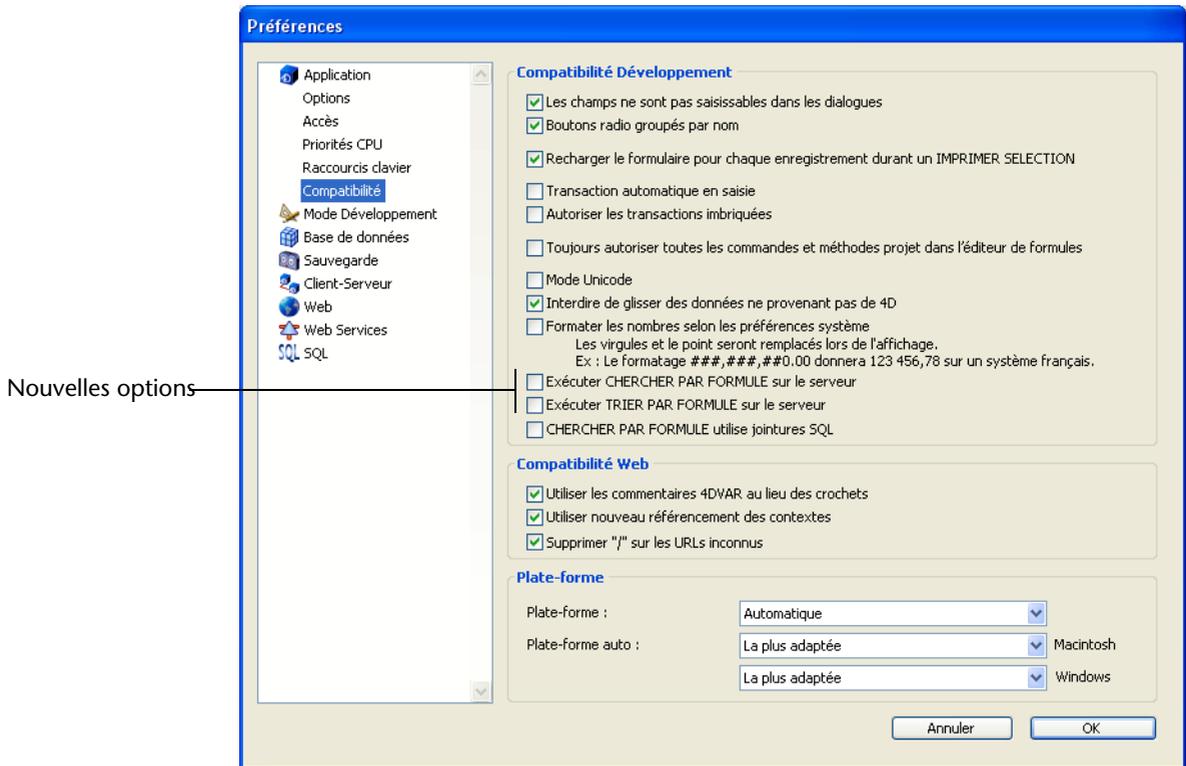
En revanche, ce principe n’est pas appliqué pour les formules utilisant des méthodes qui, elles-mêmes, font appel à des variables (la valeur des variables est évaluée sur le serveur). Dans ce contexte, il peut être judicieux d’utiliser la nouvelle option Exécuter sur serveur permettant d’exécuter une méthode sur le serveur en lui passant des paramètres (variables) (cf [paragraphe “Nouvel attribut de méthode : Exécuter sur serveur”](#), page 14).

Dans les bases de données converties en 4D v11 SQL, ce nouveau fonctionnement peut affecter les algorithmes existants. Par conséquent, par défaut dans ce contexte ces commandes continuent d’être exécutées sur le poste client.

Deux nouvelles préférences de compatibilité et deux nouveaux sélecteurs pour la commande `FIXER PARAMETRE BASE` (cf. [paragraphe “FIXER PARAMETRE BASE, Lire parametre base”](#), page 47) permettent toutefois de modifier le fonctionnement de 4D v11 dans les bases de données converties.

## Préférences (bases converties)

Deux nouvelles options disponibles dans la page "Compatibilité" permettent de modifier globalement le fonctionnement des commandes "par formule" dans les bases de données converties :



- **Exécuter CHERCHER PAR FORMULE sur le serveur** : lorsque cette option est cochée, les commandes CHERCHER PAR FORMULE et CHERCHER PAR FORMULE DANS SELECTION sont exécutées sur le serveur.
- **Exécuter TRIER PAR FORMULE sur le serveur** : lorsque cette option est cochée, la commande TRIER PAR FORMULE est exécutée sur le serveur.

## Nouveaux sélecteurs

Deux nouveaux sélecteurs (constantes) sont disponibles pour les commandes FIXER PARAMETRE BASE et Lire parametre base. Ils permettent de définir le fonctionnement par table et par process des commandes de recherche "par formule". Pour plus d'informations, reportez-vous au [paragraphe "FIXER PARAMETRE BASE, Lire parametre base"](#), page 47.

## Sélections temporaires et ensembles

Dans 4D v11 SQL, les tâches de gestion des ensembles et des sélections temporaires sont désormais réparties entre le serveur et les clients de façon transparente.

En outre, il est maintenant possible de créer des **sélections temporaires locales**, dont le nom commence par un \$. A l'image des ensembles locaux, les sélections temporaires locales sont visibles uniquement dans le process et sur le poste qui les a créées.

Le tableau suivant indique les principes de visibilité des sélections temporaires et des ensembles en fonction de leur lieu de création (le tableau est identique pour les deux types d'objets) :

Visibilité des sélections temporaires et des ensembles					
	Process client	Autres process du client	Autres clients	Process serveur	Autres process du serveur
<b>Création dans un process client</b>					
\$test	x				
test	x			x (Trigger)	
<>test	x	x			
<b>Création dans un process serveur</b>					
\$test				x	
test				x	
<>test				x	x

x = visible

## Configuration du serveur

### Numéros de ports TCP

La nouvelle version de 4D Server utilise désormais trois ports TCP pour les communications entre les serveurs internes et les clients :

- **Serveur SQL : 19812** par défaut (modifiable via la page "SQL/Configuration" des Préférences).
- **Serveur d'application : 19813** par défaut (modifiable via la page "Client-serveur/Configuration" des Préférences).
- **Serveur DB4D** (serveur de base de données) : **19814** par défaut. Ce numéro de port n'est pas modifiable directement mais il s'agit toujours du numéro de port du serveur d'application + 1.

Lorsqu'un client 4D se connecte à 4D Server, il s'adresse au port TCP du serveur d'application (19813 ou port indiqué après ':' dans l'adresse IP indiquée dans la boîte de dialogue de connexion. La connexion aux autres serveurs via leur port respectif est ensuite automatique, il n'est pas nécessaire de les préciser.

---

***Dans le cas d'accès via un routeur ou un firewall, les trois ports TCP doivent être ouverts explicitement.***

---

**Accès au Centre de sécurité et de maintenance**

Dans le cadre de l'architecture distante 4D/4D Server, la maintenance des bases peut être effectuée via le Centre de sécurité et de maintenance (CSM) ou les commandes de vérification du langage (thème "Environnement 4D").

Les possibilités d'accès à ces outils dépendent de l'application 4D, comme décrit dans le tableau suivant :

	4D Server	4D en mode distant	4D en mode local	4D Volume Desktop	Notes
<i>Accès au Centre de sécurité et de maintenance</i>					
Menu Aide/Centre de sécurité et de maintenance	Oui	-	Oui	-	
Action standard "CSM"	-	-	Oui	Oui	(1)
OUVRIER CENTRE DE SECURITE	-	-	Oui	Oui	
<i>Langage, thème "Environnement 4D"</i>					
VERIFIER FICHER DONNEES	Oui	Oui	Oui	Oui	(2)(3)
VERIFIER FICHER DONNEES OUVERT	Oui	-	Oui	Oui	(3)
Compacter fichier donnees	Oui	Oui	Oui	Oui	(2)(3)

- (1) Les menus personnalisés et les formulaires ne sont pas disponibles sur le serveur.
- (2) Ne s'applique pas au fichier de données courant
- (3) Exécution possible sur le serveur via 'Exécuter sur serveur'

# 5

## Plug-ins

Les plug-ins 4D Internet Commands et 4D View v11.2 bénéficient de nouvelles fonctions.

### 4D Internet Commands

Les commandes Internet de 4D permettent désormais d'envoyer des messages encodés en UTF-8. Pour cela, de nouveaux sélecteurs sont disponibles pour la commande `SMTP_SetPrefs`.

#### **SMTP\_SetPrefs**

`SMTP_SetPrefs (retoursLigne; typeCorps; longueurLigne) → Entier`

La commande `SMTP_SetPrefs` permet désormais l'envoi de messages utilisant le jeu de caractères UTF-8. Deux nouveaux ensembles associant un jeu de caractères (*Body-Content-Type*) et un type d'encodage (*Content-Transfer-Encoding*) peuvent être passés dans le paramètre *typeCorps* :

15 : UTF-8 & quoted-printable

16 : UTF-8 & base64

A noter que la commande `SMTP_GetPrefs` tient compte de ces modifications.

- Le code suivant enverra un message en UTF-8 encodé en quoted-printable :

```
$err:=SMTP_SetPrefs(-1;15;-1)
```

```
$err:=SMTP_QuickSend(...)
```

**Thème** : IC Send Mail.

## 4D View

Il est désormais possible de "figer" les volets d'une zone 4D View. Cette nouvelle fonction est accessible via des commandes de menu et par programmation.

### Figé et libérer les volets

La nouvelle version de 4D View permet de "figer" les volets d'une zone 4D View. Lorsqu'un volet a été figé, il est toujours conservé à l'écran, quelle que soit la navigation de l'utilisateur dans la feuille de calcul.

Une fois figé, un volet ne peut plus être redimensionné ni supprimé manuellement par l'utilisateur. Son contenu n'est plus accessible via un défilement du reste de la zone 4D View. Par exemple, si vous avez créé un volet figé incluant les trois premières lignes de la zone, il ne sera pas possible de remonter au-dessus de la 4e ligne dans le reste de la zone 4D View.

Un volet figé n'est pas verrouillé par défaut. Il reste possible de définir ou de modifier son contenu, son format, etc.

Une zone 4D View peut contenir au maximum deux volets figés : un volet horizontal en haut de la zone et un volet vertical à gauche de la zone.

Les volets figés se distinguent visuellement des volets standard par leur séparateur qui est un filet simple :

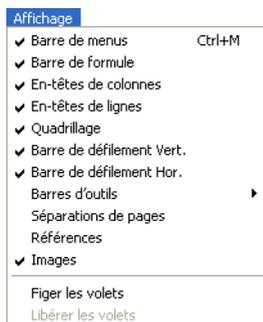
Séparateur de volet figé

	A	B	C	D	E	F
1		Salle 1	Salle 2	Salle 3	Salle 4	
2						
4	Février	516	56	45	15	
5	Mars	19	58	67	654	
6	Avril	65	56	778	15	
7	Mai	70	566	63		
8	Juin	86	56	4489	38	
9	Juillet	211	5	56	847	
10	Août	14	100	67	75	
11	Septembre	516	55	16	98	
12	Octobre	211	5	56	45	
13	Novembre	516	56	4489	15	

Il n'est pas possible de combiner des volets standard et des volets figés dans le même document. Une zone 4D View fonctionne soit en mode "volets standard", soit en mode "volets figés". Lorsqu'elle fonctionne en mode "volets figés", il n'est pas possible d'ajouter des volets. Pour pouvoir ajouter de nouveaux séparateurs, vous devez libérer les volets.

## Nouvelles commandes de menu

Deux nouvelles commandes situées dans le menu **Affichage** permettent de gérer les volets figés : **Figurer les volets** et **Libérer les volets**.



Ces commandes sont actives de façon alternée et en fonction du contexte de la zone 4D View.

*Note* Deux nouvelles commandes du langage permettent également de gérer les volets figés (reportez-vous au paragraphe ci-dessous).

- La commande **Figurer les volets** est active lorsque :
  - la zone 4D View contient un seul séparateur horizontal et/ou vertical,
  - les volets ne sont pas déjà figés.

La sélection cette commande fige le volet vertical et/ou le volet horizontal dans leur position actuelle. Il n'est plus possible d'ajouter ou de supprimer de volet dans la zone.

- La commande **Libérer les volets** est active lorsque les volets sont déjà figés dans la zone 4D View. Lorsque vous sélectionnez cette commande, les volets de la zone retrouvent leur état standard et il redevient possible d'en ajouter ou d'en supprimer.

## Nouvelles commandes

Plusieurs nouvelles commandes du langage de 4D View permettent de figer et de libérer les volets par programmation. Des commandes existantes ont également été adaptées pour prendre en charge cette nouveauté.

### PV FIGER VOLETS

PV FIGER VOLETS (zone; mode)

Paramètres	Type	Description
zone	Entier long →	Zone 4D View
mode	Entier long →	0=verrouillage du défilement, 1=verrouillage du défilement et de la modification des en-têtes

La commande PV FIGER VOLETS fige le volet horizontal et/ou le volet vertical de la *zone* 4D View. Pour que la commande fonctionne, la *zone* doit contenir au plus un volet vertical et/ou horizontal et ne pas être déjà en mode "figé".

Le paramètre *mode* permet de définir le type de verrouillage effectué dans la zone :

- Si *mode* = 0, seul le défilement à l'intérieur du volet est figé. Ce verrouillage est celui effectué lors de l'utilisation de la commande **Figier les volets** du menu **Affichage**.
- Si *mode* = 1, le verrouillage est étendu : en plus du défilement, le verrouillage affecte les modifications des en-têtes (style, taille de police...) et un clic dans une cellule du volet sélectionne la totalité de la colonne/ligne.

Si la *zone* ne contient aucun séparateur de volet ou si elle contient plus d'un volet vertical ou horizontal, la commande ne fait rien et la variable *OK* prend la valeur 0.

Si la commande est exécutée correctement, la variable *OK* prend la valeur 1 et le menu **Affichage** est modifié en conséquence : la commande **Figier les volets** est grisée et la commande **Libérer les volets** est activée.

Pour repasser la zone en mode "volets standard", il est nécessaire d'exécuter la commande PV LIBERER VOLETS ou que l'utilisateur sélectionne la commande de menu **Libérer les volets**.

- ▶ Le code suivant vérifie que la zone contient bien deux volets horizontaux (un séparateur = deux volets) et les fige en mode étendu :

```
$nbvolets:=PV Lire propriete zone(zone;pv nombre volets hor)
```

```
Si ($nbvolets=2)
```

```
    PV FIGER VOLETS(zone;1)
```

```
Fin de si
```

**Référence :** [PV LIBERER VOLETS](#)

**Thème :** PV Volet.

## PV LIBERER VOLETS PV LIBERER VOLETS (zone)

Paramètres	Type	Description
zone	Entier long →	Zone 4D View

La commande PV LIBERER VOLETS passe la zone 4D View en mode "volets standard", dans lequel il est possible d'ajouter ou de supprimer des volets, de déplacer leur séparateur, etc.

Pour que cette commande fonctionne, les volets de la zone doivent auparavant avoir été figés via la commande [PV FIGER VOLETS](#) ou la commande **Figier les volets** du menu **Affichage**.

Si la *zone* ne contient aucun séparateur ou si elle n'a pas été figée, la commande ne fait rien et la variable *OK* prend la valeur 0.

Si la commande est exécutée correctement, la variable *OK* prend la valeur 1 et le menu **Affichage** est modifié en conséquence : la commande **Libérer les volets** est grisée et la commande **Figier les volets** est activée.

- ▶ L'exemple suivant libère les volets d'une zone s'ils ont été figés :

```
PV LIRE STATUT COMMANDE(zone;pv cmd libérer les volets;$statut;  
                                                                    $coche;$nom)
```

```
Si ($statut=1)
```

```
    PV LIBERER VOLETS(zone)
```

```
Fin de si
```

**Référence :** [PV FIGER VOLETS](#)

**Thème :** PV Volet.

## Modifications du langage

Diverses modifications ont été effectuées dans le langage de 4D View pour la prise en compte de la fonction "volets figés".

### Nouveau code d'erreur

Lorsque la commande PV AJOUTER SEPARATEUR HOR ou PV AJOUTER SEPARATEUR VERT est exécutée alors que la zone est en mode "volets figés", l'erreur 92 est générée :

92 *Aucun séparateur ne peut être ajouté lorsque les volets sont figés*

### Nouvelles constantes

Les constantes suivantes ont été ajoutées dans le thème "PV Commandes" :

Constante	Type	Valeur
pv cmd figer les volets	Entier long	171
pv cmd libérer les volets	Entier long	172

### Commandes de suppression des volets

Lorsque les instructions PV SUPPRIMER SEPARATEUR HOR(zone;1) et PV SUPPRIMER SEPARATEUR VERT(zone;1) sont exécutées dans le contexte d'une zone figée, elle suppriment le volet figé et provoquent le retour de la zone en mode "volets standard".

### Commandes de gestion des propriétés de volets

Dans le contexte d'une zone en mode "volets figés", les commandes de gestion des propriétés des volets (PV FIXER PROPRIETE VOLET HOR, PV FIXER PROPRIETE VOLET VERT, PV Lire propriete volet hor et PV Lire propriete volet vert) peuvent utiliser uniquement les propriétés suivantes :

- pv volet nombre colonnes
- pv volet nombre lignes
- pv volet première colonne
- pv volet première ligne

Les autres propriétés sont inactives.

