



4^e Dimension 2004.2

ADDENDUM

Bienvenue dans 4^e Dimension version 2004.2. Ce document présente les nouveautés et modifications de cette version, regroupées dans les thèmes suivants :

- **Gestion des préférences** : les mécanismes déterminant l'emplacement des fichiers de préférences de 4D ont été modifiés et deux nouvelles préférences liées à la gestion des données ont été ajoutées.
- **Mode Structure** : des modifications ont été apportées à la gestion des objets de type listbox, des formulaires ainsi qu'à l'éditeur de menus.
- **Serveur Web** : toute méthode appelée via la balise HTML 4DVAR ou 4DVHTMLAR requiert désormais l'attribut "Disponible via 4DACTION..." .
- **Langage** : 4^e Dimension 2004.2 inclut de nouvelles commandes relatives à la navigation XML DOM ; le nouvel événement formulaire Sur après modification est désormais disponible ; les capacités des commandes existantes LISTE IMPRIMANTES, PROPRIETES PLATE FORME, Position déposer, FIXER TITRES TABLES et FIXER TITRES CHAMPS ont été étendues.
- **4D Tools** : une nouvelle option permet de régénérer les formulaires.
- **4D Write** : la fonction d'export HTML du plug-in 4D Write exploite désormais les feuilles de style CSS pour une meilleure prise en charge de HTML 4 ; en outre, deux nouvelles commandes permettent de stocker les paramètres d'impression (WR Param impression vers blob et WR BLOB VERS PARAM IMPRESSION).

Gestion des préférences

Les modifications relatives aux préférences des applications 4D 2004.2 sont de deux ordres :

- modification de l'emplacement des fichiers de préférences,
- ajout de deux nouvelles préférences.

Emplacements des fichiers de préférences

Les emplacements standard des fichiers de préférences des applications 4D (4^e Dimension, 4D Server, 4D Client, 4D Runtime, etc.) ont été modifiés en version 2004.2.

Dans les versions précédentes de 4D, ces fichiers étaient stockés dans les dossiers "All users" des dossiers de configuration utilisateur (Documents and settings\All Users\Application Data\ sous Windows et Bibliothèque:Support aux applications sous MacOS). Exception : avec 4D Client sous Windows, ces fichiers étaient stockés dans Documents and settings\NomUtilisateur\Application Data\.

Dans 4^e Dimension 2004.2, les emplacements de ces fichiers ont été unifiés. Par défaut, ce sont désormais les suivants :

	Windows	MacOS
Toutes applications hormis 4D Client	Documents and settings\All Users\Application Data\4D -> Si le dossier All Users est verrouillé : Documents and settings\NomUtilisateur\Application Data\4D	(Tous utilisateurs) Bibliothèque:Support aux applications:4D -> Si le dossier Tous utilisateurs est verrouillé : (Dossier de l'utilisateur courant):Bibliothèque:Support aux applications:4D
4D Client	Documents and settings\NomUtilisateur\Application Data\4D	(Dossier de l'utilisateur courant):Bibliothèque:Support aux applications:4D

Ce nouveau fonctionnement simplifie l'emploi de systèmes multi-sessions et permet notamment d'utiliser des applications 4D sur des systèmes où les dossiers de type "All users" sont verrouillés.

Compatibilité

Pour des raisons de compatibilité avec les dossiers de préférences existants, la procédure suivante est appliquée au lancement de l'application 4D¹ :

- L'application recherche d'abord un dossier de préférences 4D valide dans le dossier de l'utilisateur courant. S'il existe, 4D utilise ce dossier.

Note L'application se base sur la présence d'un dossier "Licenses" dans le dossier de préférences ("4D") pour déterminer sa validité.

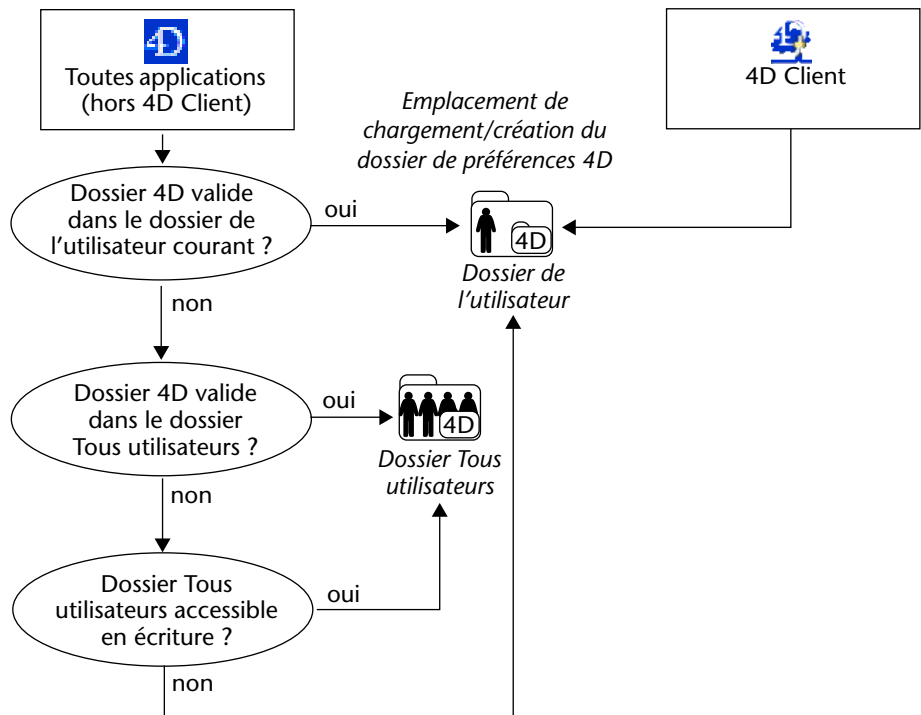
- Sinon, l'application le recherche dans le dossier "all users/tous utilisateurs" ;

1. Cette procédure concerne toutes les applications 4D hormis 4D Client. Pour cette dernière, le dossier de préférences est toujours recherché dans le dossier de l'utilisateur courant.

- s'il existe et est valide, l'application l'utilise.
- s'il n'existe pas et si le dossier "all users/tous utilisateurs" n'est pas verrouillé, l'application le crée à cet emplacement.
- si le dossier "all users/tous utilisateurs" est verrouillé, 4D crée le dossier de préférences dans le dossier de l'utilisateur courant.

Ce mécanisme vous permet notamment de contrôler l'emplacement du dossier de préférences en le déplaçant manuellement.

Le schéma suivant résume le principe de chargement/création du dossier de préférences 4D :



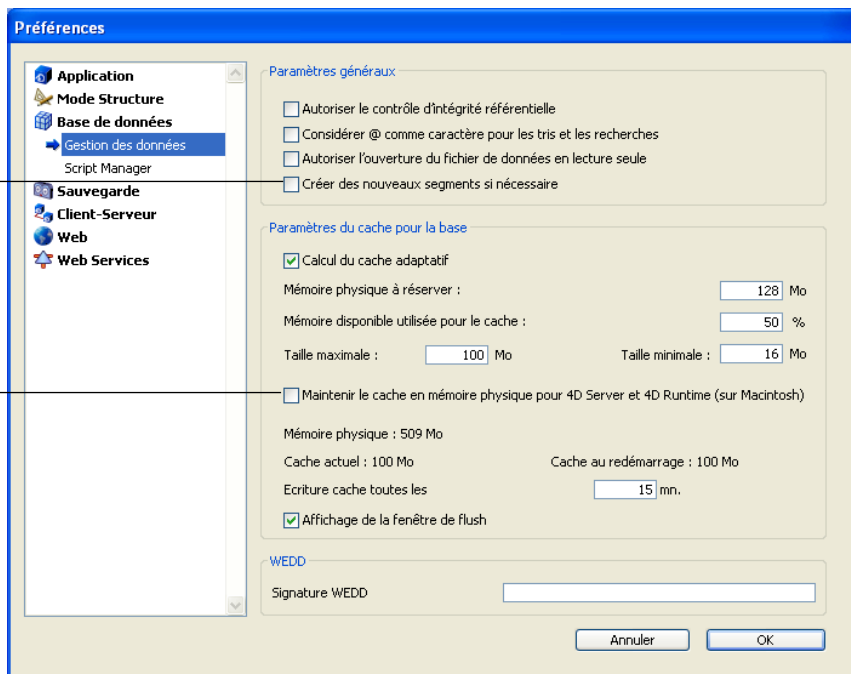
Note de compatibilité A compter de la version 2004.2, le dossier de préférences 4D situé dans le dossier du système (par exemple C:\Windows) n'est plus pris en charge. Cet emplacement restait utilisable pour des raisons de compatibilité avec les versions 6.x de 4^e Dimension. Si votre application utilisait ce dossier, vous devez déplacer manuellement son contenu vers un emplacement standard (voir ci-dessus).

Nouvelles préférences

4^e Dimension 2004.2 propose deux nouvelles préférences, disponibles dans la page **Base de données/Gestion des données** : “Créer des nouveaux segments si nécessaire” et “Maintenir le cache en mémoire physique pour 4D Server et 4D Runtime (sur Macintosh)”.

Option de création des segments de données

Option de gestion du cache sous MacOS



Créer des nouveaux segments si nécessaire

Lorsque cette option est cochée, 4^e Dimension crée automatiquement de nouveaux segments de données lorsque la taille maximale du fichier de données ou de ses segments est atteinte pendant l’exploitation de la base. Aucune boîte de dialogue n’est affichée. La taille maximale est de 2 Go pour le fichier de données et 2 Go par défaut pour chaque segment, dans la limite de 128 Go.

Chaque segment automatique est d’une taille maximale de 2 Go et est stocké à côté du dernier segment créé ou du fichier de données. Son nom est basé sur celui du fichier de données, conformément aux principes standard pour les segments : pour un fichier de données nommé MonData.4DD, le premier segment est nommé MonData.4DS, le deuxième segment MonData2.4DS et ainsi de suite.

Cette option n’empêche pas la création manuelle de segments. Les segments créés automatiquement apparaîtront dans la boîte de dialogue de gestion des segments en mode Utilisation.

Si un segment automatique ne peut être créé par manque de place sur le disque de destination, une boîte de dialogue d'alerte est affichée. Il est dans ce cas nécessaire de procéder à la création manuelle du nouveau segment à un autre emplacement.

Maintenir le cache en mémoire physique pour 4D Server et 4D Runtime (sur Macintosh)

Cette option avancée permet de “forcer” le cache à rester dans la mémoire physique (mémoire RAM) de la machine lorsque la base est exécutée par 4D Runtime (toutes versions) ou 4D Server sous MacOS.

Sous MacOS en effet, par défaut le cache peut être déchargé en partie ou en totalité dans la mémoire virtuelle de la machine en cas de besoin (la mémoire virtuelle est une zone réservée sur le disque dur). Ce mécanisme pouvant dans certains cas altérer sensiblement les performances des applications déployées avec 4D Server ou 4D Runtime, vous pouvez cocher cette option afin que le cache soit maintenu en permanence dans la mémoire physique. Bien entendu, la configuration de la machine (taille de la mémoire RAM, nombre d'applications exécutées simultanément...) doit être compatible avec ce principe.

Note Cette option n'affecte pas le fonctionnement des applications exécutées avec l'application 4^e Dimension monoposte.

Mode Structure

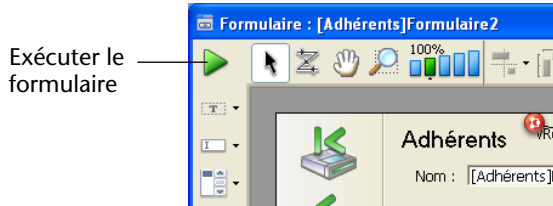
Les modifications apportées dans le mode Structure de 4^e Dimension 2004.2 concernent les formulaires, les list box ainsi que l'éditeur de menus.

Formulaires

Test des formulaires

Vous pouvez désormais tester l'exécution d'un formulaire directement depuis l'éditeur de formulaires en mode Structure — à l'image de l'exécution d'une méthode projet depuis l'éditeur de méthodes.

Il vous suffit pour cela de cliquer sur le bouton d'exécution situé en haut à gauche de la fenêtre de l'éditeur de formulaires :



Note Ce bouton n'est pas disponible pour les formulaires destinés à l'impression (types Formulaire impression détaillé et Formulaire impression liste).

Lorsque vous cliquez sur ce bouton, 4^e Dimension bascule automatiquement en mode Utilisation et affiche le formulaire dans son contexte (liste d'enregistrements pour un formulaire liste et enregistrement courant en page pour un formulaire détaillé). Le formulaire est exécuté dans le process principal.

Note L'icône de verrouillage (en forme de cadenas) a été déplacée en haut à droite de la fenêtre.

Avec contraintes de redimensionnement

Une nouvelle option de compatibilité est disponible dans la Liste des propriétés (thème "Taille de la fenêtre") pour les formulaires des bases de données 2003.x ou antérieures converties en version 2004.2 : **Avec contraintes**.

Cette option permet de reproduire le fonctionnement de l'ancienne propriété de formulaire "Redimensionnable" : lorsque cette option n'est pas cochée, les mécanismes prenant en charge le redimensionnement des objets et les contraintes de taille des fenêtres sont désactivés. Dans ce mode, l'utilisateur redimensionne librement la fenêtre du formulaire mais les objets qu'elle contient ne sont ni redimensionnés ni déplacés. Les propriétés de taille mini/maxi ou fixe ainsi que les propriétés de redimensionnement des objets sont ignorées.

Ce fonctionnement correspond aux anciennes versions de 4^e Dimension et ne doit être utilisé que par souci de compatibilité dans le cadre d'interfaces spécifiques. Le fonctionnement des bases 4D créées en version 2004 correspond à l'option cochée (mode standard).

Glisser-déposer dans les list box

Il est désormais possible d'effectuer un glisser-déposer de ligne de list box vers une autre list box ou un autre objet 4D et inversement.

Les principes suivants sont mis en oeuvre :

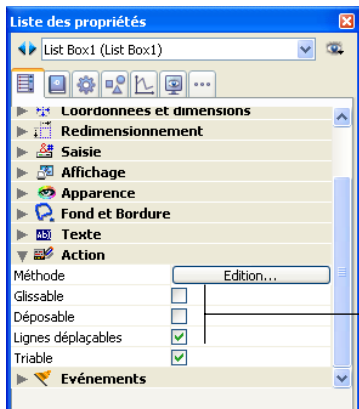
- seules les lignes des list box sont concernées ; il n'est pas possible de glisser-déposer des colonnes (il reste toutefois possible de déplacer des colonnes à l'intérieur d'une même list box) ;
- seule une ligne unique peut être glissée-déposée ; le glisser-déposer d'une sélection de plusieurs lignes n'est pour l'instant pas possible ;
- la gestion du glisser-déposer dans les list box s'effectue à l'aide des mécanismes de glisser-déposer standard de 4^e Dimension :
 - propriétés "Glissable" et "Déposable" pour l'objet list box dans l'éditeur de formulaires,
 - événements formulaire Sur glisser et Sur déposer,
 - commandes PROPRIETES GLISSER DEPOSER et Position déposer. La commande Position déposer admet désormais un nouveau paramètre, permettant de connaître le numéro de la colonne sur laquelle le déposer a été effectué. Pour plus d'informations sur ce point, reportez-vous au [paragraphe "Position déposer", page 13](#).

Note La programmation du glisser-déposer doit être effectuée dans la méthode objet de la list box.

Pour plus d'informations sur le glisser-déposer dans 4^e Dimension, reportez-vous au chapitre "Glisser-Déposer" dans le manuel *Langage* de 4^e Dimension.

■ Interaction entre les propriétés Glissable, Déposable et Lignes déplaçables

Ces trois propriétés peuvent être cochées pour une list box :

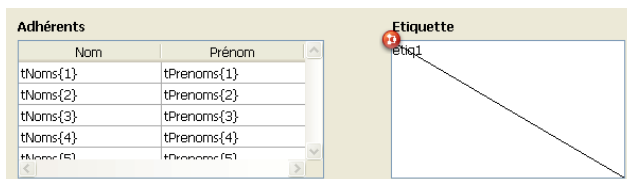


Propriétés en interaction

La combinaison de ces propriétés fournit les résultats suivants :

	Combinaison 1	Combinaison 2	Combinaison 3
Glissable	-	✓	✓
Déposable	-	✓ ou -	✓ ou -
Lignes déplaçables	✓	✓	-
Commentaire	Le fonctionnement de la list box est identique à celui des versions précédentes de 4 ^e Dimension. Pour des raisons de compatibilité, cette configuration est utilisée par défaut.	L'option "Ligne déplaçable" est prioritaire en cas de déplacement d'une ligne. Le glisser n'est donc pas possible. Si l'option "Déposable" est cochée, la list box peut accepter le déposer d'objets.	Le glisser-déposer est activé. Si l'option "Déposable" est cochée, il est également possible de gérer le glisser-déposer à l'intérieur de la list box.

- ▼ Nous souhaitons remplir une zone de texte (par exemple une étiquette) avec des données glissées depuis une list box.



Le contenu de la list box est généré à partir de la table [Adhérents].

- les propriétés d'événements Sur glisser et Sur déposer ont été cochées pour le formulaire,
- la propriété d'événement Sur déposer a été cochée pour la zone de texte.

Voici la méthode de l'objet *etiql* :

Au cas ou

:(Evenement formulaire=Sur glisser)

PROPRIETES GLISSER DEPOSER(\$source;\$lignetab;\$numprocess)

Si (\$source=Pointeur vers("list box1"))

\$0:=0 ` on accepte le glisser

Sinon

\$0:=-1

Fin de si

:(Evenement formulaire=Sur déposer)

PROPRIETES GLISSER DEPOSER(\$source;\$lignetab;\$numprocess)

CHERCHER([Adhérents];[Adhérents]Nom=tNoms{\$lignetab})

Si (Enregistrements trouves([Adhérents])#0)

etiql:=[Adhérents]Nom+" "+[Adhérents]Prénom+

Caractere(Retour chariot)+[Adhérents]Adresse+

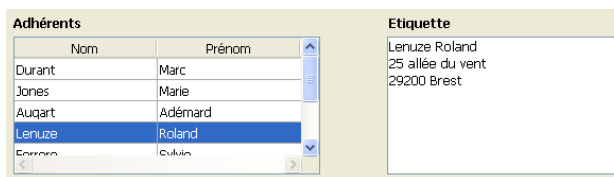
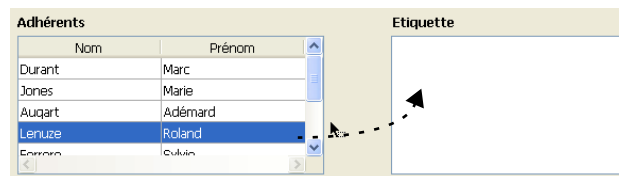
Caractere(Retour chariot)+[Adhérents]Code postal+

" "+[Adhérents]Ville

Fin de si

Fin de cas

Il est dès lors possible d'effectuer l'action suivante :

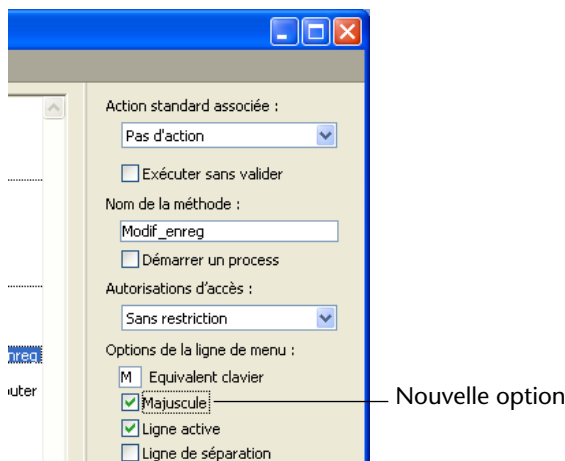


- ▼ Reportez-vous à la description de la commande **Position déposer** pour un exemple de déposer dans une list box.

Note L'ergonomie du passage en saisie dans les cellules de list box a été modifiée dans 4^e Dimension 2004.2. Ce passage, effectué via deux clics espacés par un intervalle long, est maintenant plus aisé.

Touche Majuscule pour les équivalents clavier

Il est désormais possible d'ajouter la touche **Majuscule** à un équivalent clavier associé à une commande de menus personnalisée. Pour cela, il suffit de cocher l'option "Majuscule" dans l'éditeur de menus :



Cette nouveauté double le nombre d'équivalents clavier utilisables dans les barres de menus créés. A noter qu'un équivalent clavier est toujours associé à la touche **Ctrl** (Windows) ou **Commande** (MacOS). Lorsque l'option "Majuscule" est cochée, l'équivalent clavier défini est du type **Ctrl+Maj+lettre** (Windows) ou **Commande+Maj+lettre** (MacOS).

Prise en charge des alphabets non romans

La prise en charge des langues à alphabet non roman a été étendue dans 4^e Dimension 2004.2.

Les scripts suivants sont désormais fournis avec les applications 4^e Dimension : arabe, chinois simplifié, chinois traditionnel, croate, cyrillique, europe de l'est, grec, hébreu, islandais, japonais, coréen, roumain, thaïlandais, européen-US et vietnamien. Ces scripts donnent accès à une soixantaine de langues.

4^e Dimension utilise automatiquement le script correspondant à la langue courante du système pour la saisie, l'affichage et les traitements des données (tels que les tris).

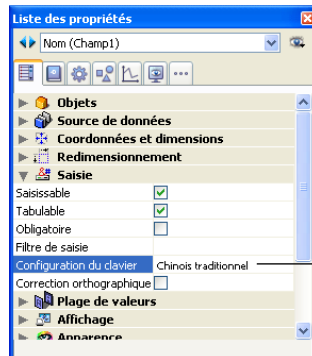
Note Pour plus de précisions sur l'étendue de la compatibilité de 4^e Dimension avec chacune de ces langues, veuillez consulter les matrices de certification des langues qui seront publiées par 4D en fonction des tests effectués (<http://www.4d.fr>).

Propriété "Configuration du clavier"

Dans les formulaires, il est désormais possible d'assigner une propriété de **Configuration du clavier** spécifique aux objets saisissables suivants :

- champs
- variables saisissables
- listes hiérarchiques
- list box
- combo box.

Cette propriété, accessible dans la Liste des propriétés, permet la saisie et l'affichage de données en utilisant un script de langue différent de celui du système.



Propriété Configuration du clavier

Cette propriété "force" le script à utiliser pour la conversion en interne des caractères saisis. Elle provoque en outre le changement automatique de clavier lorsque l'objet reçoit le curseur.

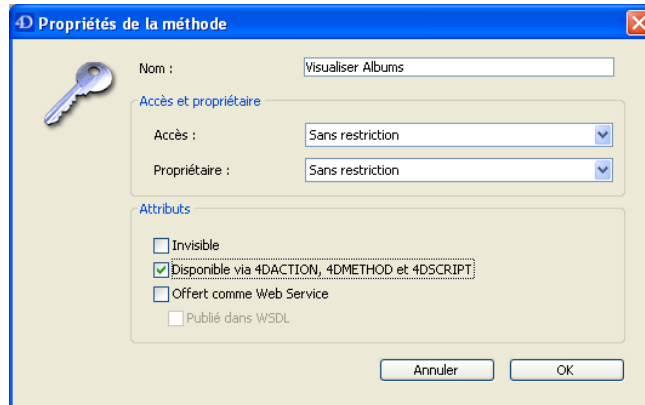
Si l'utilisateur change le clavier courant au cours de la session 4D sans que cette propriété ait été définie, la saisie dans les formulaires sera possible mais les caractères ne seront pas correctement stockés (ils disparaissent au moment où le curseur sort de la zone).

IMPORTANT : Cette propriété n'affecte que la saisie et l'affichage des données. Les traitements (tris, recherche, etc.) sont toujours effectués avec le script courant du système.

Serveur Web

Extension de l'attribut "Disponible via 4DACTION..."

Le mécanisme de sécurité nécessitant que la propriété "Disponible via 4DACTION, 4DMETHOD et 4DSCRIPT" soit cochée pour les méthodes appelées par des requêtes HTTP a été étendu aux balises 4DVAR et 4DHTMLVAR.



Dans les versions précédentes de 4^e Dimension, seule la balise 4DSCRIPT était soumise à cette règle (en plus des URLs 4DACTION et 4DMETHOD).

A compter de 4^e Dimension 2004.2, cette option doit également être cochée pour les méthodes appelées via les balises 4DVAR et 4DHTMLVAR.

Note Il n'est pas nécessaire de déclarer explicitement les sous-méthodes appelées par les méthodes exécutées via ces balises.

Langage

Les modifications suivantes ont été apportées au langage de 4^e Dimension :

- la commande Position déposer accepte désormais un paramètre,
- la commande PROPRIETES PLATE FORME retourne désormais la langue courante du système,
- la commande LISTE IMPRIMANTE retourne des informations supplémentaires sous MacOS,

- trois nouvelles commandes de navigation DOM ont été ajoutées dans le thème “XML”,
- le nouvel événement formulaire Sur après modification permet de gérer toute modification effectuée dans une zone de saisie, y compris les événements d’édition tels que couper ou coller,
- les commandes FIXER PARAMETRE BASE et Lire parametre base admettent un nouveau sélecteur.

Position déposer

Position déposer $\{(numColonne)\}$ → Numérique

Paramètres	Type	Description
<i>numColonne</i>	Entier long	← Numéro de colonne ou -1 si le déposer a lieu après la dernière colonne
Résultat	Numérique	← Numéro (tableau/list box) ou position (liste) de l’élément de destination ou -1 si le déposer a lieu après le dernier élément

Thème : Glisser-Déposer

La commande Position déposer admet désormais le paramètre *numColonne*, facultatif, retournant le numéro de la colonne sur laquelle le déposer a eu lieu. Cette commande est destinée à la gestion du déposer dans les objets de type list box. Pour plus d’informations sur ce point, reportez-vous au [paragraphe “Glisser-déposer dans les list box”, page 7.](#)

- ▼ Dans l’exemple suivant, une liste de sommes doit être ventilée par mois et par personne. L’opération s’effectue par glisser-déposer depuis une zone de défilement :

Mois	Jean	Marc	Pierre	Sommes versées
Janvier	5 254	272	0	31 901
Février	870	2 782	873	11 721
Mars	572	3 324	787	31 008
Avril	0	0	0	19 341
Mai	198	0	0	7 675
Juin	0	771	337	26 494
Juillet	0	0	0	544
Août	0	0	0	16 979
Septembre	0	0	0	27 642
Octobre	0	0	0	4 750
Novembre	0	0	0	22 509
Décembre	0	0	0	10 758

Zone de défilement

List box

La méthode objet de la list box contient le code suivant :

Au cas ou

:(Evenement formulaire=Sur glisser)

PROPRIETES GLISSER DEPOSER(\$source;\$lignetab; \$numprocess)

Si (\$source=Pointeur vers("ZD1")) `Si le déposer provient bien de la
`zone de défilement

\$0:=0

Sinon

\$0:=-1 `On refuse le déposer

Fin de si

:(Evenement formulaire=Sur déposer)

PROPRIETES GLISSER DEPOSER(\$source;\$lignetab;\$numprocess)

\$numligne:=**Position déposer**(\$numcol)

Si (\$numcol=1)

BEEP

Sinon

Au cas ou `Addition des valeurs déposées

: \$numcol=2

Jean{\$numligne}:=Jean{\$numligne}+ZD1{\$lignetab}

: \$numcol=3

Marc{\$numligne}:=Marc{\$numligne}+ZD1{\$lignetab}

: \$numcol=4

Pierre{\$numligne}:=Pierre{\$numligne}+ZD1{\$lignetab}

Fin de cas

SUPPRIMER LIGNES(ZD1;\$lignetab) `Mise à jour de la zone

Fin de si

Fin de cas

PROPRIETES PLATE FORME

PROPRIETES PLATE FORME (plateForme{; système{; machine{; *langue*{}})

Paramètres	Type	Description
plateForme	Numérique	← 1 = Macintosh 68K (<i>obsolète</i>), 2 = Power Macintosh, 3 = Windows
système	Numérique	← Dépend de la version que vous utilisez
machine	Numérique	← Dépend de la version que vous utilisez
<i>langue</i>	<i>Numérique</i>	← <i>Dépend du système que vous utilisez</i>

Thème : Environnement système

Le nouveau paramètre *langue*, facultatif, permet de connaître la langue courante du système sur lequel est exécutée la base.

Voici la liste des codes pouvant être retournés dans ce paramètre, ainsi que leur signification :

Codes	Langues
1	Arabe
2	Bulgare
3	Catalan
4	Chinois
5	Tcheque
6	Danois
7	Allemand
8	Grec
9	Anglais
10	Espagnol
11	Finlandais
12	Français
13	Hébreu
14	Hongrois
15	Islandais
16	Italien
17	Japonais
18	Coréen
19	Néerlandais
20	Norvégien
21	Polonais
22	Portugais
24	Roumain
25	Russe
26	Croate
26	Serbe
27	Slovaque
28	Albanais
29	Suédois
30	Thailandais
31	Turc
33	Indonésien
34	Ukrainien
35	Bélarusse
36	Slovène

Codes	Langues
37	Estonien
38	Letton
39	Lithuanien
41	Farsi
42	Vietnamien
45	Basque
54	Afrikaans
56	Féroïen

Note Si la commande n'a pas pu identifier la langue du système, la valeur 9 (Anglais) est retournée.

LISTE IMPRIMANTES LISTE IMPRIMANTES (tabNoms; tabNomsAlt; tabModèles}}

Sous MacOS, le deuxième paramètre de cette commande retourne désormais pour chaque imprimante son nom personnalisé, modifiable par l'utilisateur. Ce nom peut être utilisé par exemple dans des boîtes de dialogue.

Nouvelles commandes DOM

Le thème de commandes XML s'est enrichi de trois nouvelles commandes destinées à compléter les possibilités de navigation DOM au sein des structures XML.

A noter que deux commandes DOM existantes du thème XML ont été renommées (cf. [paragraphe "Commandes DOM renommées", page 18](#)).

DOM Lire element XML parent

DOM Lire element XML parent (refElément{; nomElémentPar{; valeurElémentPar}}) → Chaîne

Paramètres	Type	Description
refElément	Chaîne	→ Référence d'élément XML
nomElémentPar	Chaîne	← Nom de l'élément XML parent
valeurElémentPar	Chaîne	← Valeur de l'élément XML parent
Résultat	Chaîne	← Référence de l'élément XML parent (16 caractères)

La commande DOM Lire element XML parent retourne une référence XML vers le "parent" de l'élément XML passé en référence dans

refElément. Cette référence pourra être utilisée avec les autres commandes d'analyse XML.

Les paramètres *nomElémentPar* et *valeurElémentPar*, s'ils sont passés, reçoivent respectivement le nom et la valeur de l'élément parent.

Si la commande a été correctement exécutée, la variable système OK prend la valeur 1. Sinon, elle prend la valeur 0.

DOM Lire element XML frere precedent

DOM Lire element XML frere precedent (*refElément*{; *nomElémentFrère*{; *valeurElémentFrère*}}) → Chaîne

Paramètres	Type	Description
<i>refElément</i>	Chaîne	→ Référence d'élément XML
<i>nomElémentFrère</i>	Chaîne	← Nom de l'élément XML frère
<i>valeurElémentFrère</i>	Chaîne	← Valeur de l'élément XML frère
Résultat	Chaîne	← Référence de l'élément XML frère (16 caractères)

La commande DOM Lire element XML precedent retourne une référence vers le précédent "frère" de l'élément XML passé en référence. Cette référence pourra être utilisée avec les autres commandes d'analyse XML.

Les paramètres *nomElémentFrère* et *valeurElémentFrère*, s'ils sont passés, reçoivent respectivement le nom et la valeur de l'élément "frère" précédent.

Cette commande permet de naviguer parmi les "enfants" d'un élément XML.

Avant le premier "frère", la variable système OK prend la valeur 0.

Si la commande a été correctement exécutée et si l'élément référencé n'est pas le premier "enfant" de la structure, la variable système OK prend la valeur 1. Si une erreur se produit ou si l'élément analysé est le premier "enfant" de la structure, elle prend la valeur 0.

DOM Lire dernier element XML enfant DOM Lire dernier element XML enfant(refElément{; nomElémentEnf{; valeurElémentEnf}) → Chaîne

Paramètres	Type	Description
refElément	Chaîne	→ Référence d'élément XML
nomElémentEnf	Chaîne	← Nom de l'élément enfant
valeurElémentEnf	Chaîne	← Valeur de l'élément enfant
Résultat	Chaîne	← Référence de l'élément XML (16 caractères)

La commande DOM Lire dernier element XML retourne une référence XML vers le dernier "enfant" de l'élément XML passé en référence dans *refElément*. Cette référence pourra être utilisée avec les autres commandes d'analyse XML.

Les paramètres *nomElémentEnf* et *valeurElémentEnf*, s'ils sont passés, reçoivent respectivement le nom et la valeur de l'élément enfant.

Si la commande a été correctement exécutée, la variable système OK prend la valeur 1. Sinon, elle prend la valeur 0.

- ▼ Récupération de la référence du dernier élément XML du parent racine. La structure XML (C:\import.xml) est préalablement chargée dans un BLOB :

```
C_BLOB(maVarBlob)
C_ALPHA(16;$ref_XML_Parent;$ref_XML_Enfant)
C_TEXTE($nomEnfant;$valeurEnfant)
```

```
DOCUMENT VERS BLOB("c:\import.xml";maVarBlob)
$ref_XML_Parent:=DOM Analyser variable XML(maVarBlob)
$ref_XML_Enfant:=DOM Lire dernier element XML enfant
($ref_XML_Parent;$nomEnfant;$valeurEnfant)
```

Commandes DOM renommées

Pour des raisons de cohérence et de clarté, deux commandes XML existantes ont été renommées dans 4D 2004.2 :

Nom précédent	Nouveau nom 4D 2004.2
DOM Lire premier element XML	DOM Lire premier element XML enfant
DOM Lire element XML suivant	DOM Lire element XML frere suivant

Nouvel événement formulaire Sur après modification

Un nouvel événement formulaire est disponible dans 4^e Dimension 2004.2 : Sur après modification. Lorsqu'il est utilisé, cet événement est généré après chaque modification du contenu d'une zone saisissable (champ ou variable), quelle que soit l'action à l'origine de cette modification. Tous les types de données "saisissables" sont concernés : alphas, textes, dates, heures, numériques et images.

Les actions pouvant déclencher cet événement sont :

- les actions d'édition standard entraînant une modification du contenu telles que *coller*, *couper*, *effacer* ou *annuler* ;
- le déposer d'une valeur (action similaire au *coller*) ;
- toute saisie au clavier effectuée par l'utilisateur ; dans ce cas, l'événement Sur après modification est généré après les événements Sur avant frappe clavier et Sur après frappe clavier s'ils sont utilisés.
- une modification effectuée via une commande du langage simulant une action utilisateur (i.e. GENERER FRAPPE CLAVIER ou GENERER EVENEMENT).

Attention, les actions suivantes ne déclenchent PAS cet événement :

- les actions d'édition ne modifiant pas le contenu de la zone, comme *copier* ou tout *sélectionner* ;
- le glisser d'une valeur (action similaire au *copier*) ;
- les modifications de contenu effectuées par programmation, à l'exception des commandes citées précédemment.

Cet événement permet de contrôler les actions utilisateur afin, par exemple, d'interdire de coller un texte trop volumineux, de filtrer certains caractères ou d'empêcher le *couper* d'un champ de mot de passe.

Note Dans la plupart des cas, il est désormais conseillé d'utiliser ce nouvel événement plutôt que l'événement existant Sur après frappe clavier, plus restrictif.

FIXER PARAMETRE BASE, Lire parametre base

FIXER PARAMETRE BASE({table;} sélecteur; valeur)

Lire parametre base({table;} sélecteur) → Entier long

Une nouvelle constante de *sélecteur* est disponible :

Sélecteur	Valeur	Portée
Précision affichage réels	32	Application 4D

■ Sélecteur = 32 (Précision affichage réels)

Description : Nombre de chiffres non significatifs tronqués à partir de la droite par l'algorithme d'affichage à l'écran des réels. Cette valeur est définie pour l'application et la session courantes.

Par défaut, la valeur de cette option est 4. La valeur 0 indique que la valeur par défaut est utilisée et que le paramètre n'a pas été modifié durant la session.

Pour des raisons historiques, 4^e Dimension travaille avec des nombres réels stockés sur 10 octets et les convertit sur 8 octets lors de leur traitement. Ce principe est entièrement transparent et n'affecte pas les calculs, toutefois certains résultats peuvent ne pas donner à l'affichage le résultat escompté. Par exemple, l'opération 4,1-4,09 affiche comme résultat 0,0099999999999999780000, mais la recherche de 0,01 retrouve la bonne valeur.

Voici comment 4D procède pour afficher un réel : prenons par exemple la valeur 8,974999999999996158 obtenue par un calcul (la valeur 8,975 étant normalement le résultat attendu). L'algorithme permettant d'arrondir au plus juste enlève par défaut les 4 derniers chiffres (6158) puis vérifie si le dernier chiffre restant est un 0 ou un 9. Si c'est 0, l'algorithme remonte jusqu'au premier 0 et supprime tous les 0. Si la valeur est 9, l'algorithme remonte jusqu'au premier 9 et arrondit la partie décimale à la valeur supérieure. Dans notre exemple, la valeur 8,974999999999996158 se transforme donc en 8,975.

Il peut arriver que certains résultats finissent par 5 chiffres non significatifs, comme par exemple 8,9749999999999986158. Dans ce cas, l'algorithme ne pourra pas arrondir correctement la valeur car il trouvera 8 comme valeur et ne fera alors rien.

Vous pouvez souhaiter que l'algorithme de précision tronque plus ou moins de chiffres en fonction des spécificités de votre base. Dans ce cas, passez une *valeur* personnalisée. Hormis pour le zéro (choix de la valeur interne de 4D), cette valeur indiquera le nombre de chiffres tronqués par l'algorithme de précision.

A noter que ce paramétrage n'influe que sur l'affichage des numériques, non sur leur traitement interne.

Note Dans les versions précédentes de 4^e Dimension, ce paramétrage était disponible via l'utilitaire 4D Customizer Plus.

FIXER TITRES CHAMPS, FIXER TITRES TABLES

FIXER TITRES CHAMPS (table | sous-table; titresChamps; numChamps; *)

Paramètres	Type	Description
table sous-table	Table Sous-table	→ Table ou sous-table à redéfinir
titresChamps	Tab Alpha	→ Nouveaux titres des champs
numChamps	Tab Num	→ Numéros des champs
*	*	→ <ul style="list-style-type: none"> • Si passé = utiliser la structure virtuelle dans l'éditeur de formules • Si omis = ne pas l'utiliser

FIXER TITRES TABLES (titresTables; numTables; *)

Paramètres	Type	Description
titresTables	Tab Alpha	→ Nouveaux noms des tables
numTables	Tab Num	→ Numéros des tables
*	*	→ <ul style="list-style-type: none"> • Si passé = utiliser la structure virtuelle dans l'éditeur de formules • Si omis = ne pas l'utiliser

Les commandes FIXER TITRE CHAMPS et FIXER TITRES TABLES acceptent désormais le paramètre facultatif * supplémentaire.

Si ce paramètre est omis, les formules exécutées dans 4D ne pourront pas utiliser les noms personnalisés définis par ces commandes (nouveau fonctionnement par défaut dans 4^e Dimension 2004.2, identique au fonctionnement des versions 2003.x).

S'il est passé, les noms définis par ces commandes pourront être utilisés dans les formules exécutées par 4^e Dimension (fonctionnement par défaut dans 4^e Dimension 2004.0 et 2004.1). Dans ce cas, les noms personnalisés ne doivent pas contenir de caractères "interdits" par l'interpréteur du langage de 4D (tels que -?!).

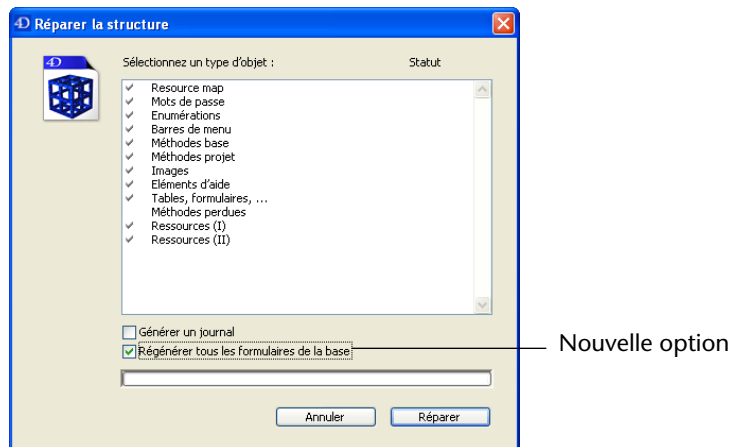
Note de compatibilité : La version 2004 de 4D a introduit un nouvel éditeur de formules tenant compte des noms de tables et de champs personnalisés ("structure virtuelle") définis à l'aide des commandes FIXER TITRES CHAMPS et FIXER TITRES TABLES. Cette nouveauté a entraîné l'instauration de règles plus contraignantes pour la définition

des noms personnalisés, car ils doivent être compatibles avec l'interpréteur du langage de 4D. Afin de permettre aux développeurs de s'affranchir de cette contrainte, le "nouveau" mode doit maintenant être explicitement activé via le paramètre *. Par défaut désormais, il n'est pas possible d'utiliser d'élément de la structure virtuelle dans les formules, comme dans les versions 2003.x de 4^e Dimension.

4D Tools : Mise à jour des formulaires

Suite à la mise à jour d'une base de données avec une nouvelle version de 4^e Dimension, certains formulaires peuvent présenter des anomalies de fonctionnement, notamment lorsque la base est ancienne.

Pour remédier à ce problème, une nouvelle option est disponible dans la fenêtre de réparation de la structure de 4D Tools : **Régénérer tous les formulaires de la base.**



Note Pour afficher cette fenêtre, cliquez sur le bouton **Réparer** dans la page "Structure".

Cette option permet de mettre à jour et de réenregistrer tous les formulaires de la base en une seule opération. Lorsqu'elle est cochée, 4^e Dimension "force" l'analyse et l'enregistrement des formulaires au cours de la procédure de réparation.

Lorsque la base d'origine contient des formulaires endommagés, l'utilisation de cette option peut entraîner des effets indésirables. Il est donc recommandé de l'utiliser sur une copie de la base, que vous aurez pris soin de compacter au préalable avec 4D Tools.

4D Write

La version 2004.2 de 4D Write comporte les nouveautés suivantes :

- Amélioration de la conversion des documents 4D Write en HTML.
- Nouvelles commandes de gestion des paramètres d'impression.

Conversion des documents en HTML

La conversion des documents 4D Write en HTML a été enrichie. 4D Write utilise désormais les CSS afin de convertir les styles et les formats de manière plus proche du document original.

Le tableau ci-dessous précise les éléments désormais pris en charge :

Attributs	Conversion HTML
Style Normal	Oui
Style Gras	Oui
Style Italique	Oui
Style Barré	Oui
Style Exposant	Oui
Style Indice	Oui
Style Ombré	Non
Soulignement	Oui (soulignement simple) - les variantes de soulignement sont converties en soulignements simples
Police	Oui
Taille de police	Oui
Couleurs de premier plan et d'arrière-plan	Oui (textes et des bordures) - les couleurs des textes barrés, ombrés et des soulignements ne sont pas converties.

Puce	Oui (puces "carré noir" et "cercle blanc") - les autres types de puces sont remplacés par des puces standard
Alignement	Oui
Interlignage	Oui
Indentation	Oui
Bordure	Oui
Tabulations	Oui, remplacées par des espaces (nbsp)
Image dans le texte	Oui (QuickTime doit être installé sur la machine effectuant la conversion)
Image sur le fond	Non
Lien hypertexte	Oui (liens standard) - les liens provoquant l'ouverture d'un document ou l'exécution d'une méthode 4D ne sont pas convertis.

Nouvelles commandes d'impression

Deux nouvelles commandes permettent de compléter les capacités de 4D Write à gérer par programmation des paramètres d'impression : WR Param impression vers blob et WR BLOB VERS PARAM IMPRESSION (thème "WR Impression").

WR Param impression vers blob

WR Param impression vers blob (zone) → BLOB

Paramètres	Type	Description
zone	Entier long	→ Zone 4D Write
Résultat	BLOB	← BLOB contenant les paramètres d'impression

La commande WR Param impression vers blob stocke dans un BLOB les paramètres d'impression courants de la zone 4D Write. La zone peut être une fenêtre externe, une zone incluse ou une zone hors écran.

Le BLOB stocke tous les paramètres utilisés pour l'impression :

- les paramètres de configuration (papier, orientation, échelle) ;
- les paramètres d'impression proprement dit (nombre de copies, alimentation, etc.).

Note Sous Windows, les paramètres stockés dans le BLOB incluent l'imprimante.

Cette commande permet de sauvegarder les paramètres d'impression de la zone 4D Write, quels que soient le modèle d'imprimante et les paramètres d'impression accessibles. Le BLOB retourné ne doit pas être modifié par programmation, il ne peut être exploité que par la commande WR BLOB VERS PARAM IMPRESSION (ou la commande 4D Pack AP BLOB to print settings).

La commande WR Param impression vers blob peut être utilisée par exemple pour sauvegarder les paramètres d'impression courants avant de modifier temporairement une option à l'aide de la commande WR FIXER OPTION IMPRESSION. Une fois l'impression effectuée, la commande WR BLOB VERS PARAM IMPRESSION permet de rétablir les paramètres courants.

La variable système OK prend la valeur 1 si le BLOB a été correctement généré et 0 dans le cas contraire. L'erreur 1014 est générée si aucune imprimante n'est sélectionnée.

Référence : WR BLOB VERS PARAM IMPRESSION

WR BLOB VERS PARAM IMPRESSION

WR BLOB VERS PARAM IMPRESSION(zone;paramImpression{;typeParam})

Paramètres	Type	Description
zone	Entier long	→ Zone 4D Write
paramImpression	BLOB	→ BLOB contenant les paramètres d'impression
typeParam	Entier long	→ 0 = configuration et impression, 1 = impression

La commande WR BLOB VERS PARAM IMPRESSION remplace les paramètres d'impression courants de la zone 4D Write par ceux contenus dans le BLOB *paramImpression*.

La *zone* peut être une fenêtre externe, une zone incluse ou une zone hors écran. Toutefois, du fait des mécanismes de gestion des paramètres d'impression de 4D Write, cette commande ne permet pas d'adresser toutes les zones en passant 0 dans le paramètre *zone*.

Le BLOB *paramImpression* doit avoir été généré par la commande WR Param impression vers blob (ou la commande 4D Pack AP Print settings to BLOB).

paramImpression contient deux types de paramètres :

- les paramètres de configuration (papier, orientation, échelle) ;
- les paramètres d'impression proprement dit (nombre de copies, alimentation, etc.).

Note Sous Windows, les paramètres stockés dans le BLOB incluent l'imprimante.

Si vous passez 0 dans *typeParam* ou omettez ce paramètre, tous les paramètres sont utilisés. Si vous passez 1, seuls les paramètres d'impression sont utilisés ; les paramètres de configuration d'impression sont inchangés.

Les nouveaux paramètres d'impression sont appliqués au document présent dans la *zone*.

Note Les paramètres d'impression ne sont pas formatés de la même manière sous Windows et MacOS. Par conséquent, la compatibilité du BLOB *paramImpression* entre les deux plates-formes n'est pas garantie.

La variable système OK prend la valeur 1 si le BLOB a été correctement chargé et 0 dans le cas contraire.

Si aucune imprimante n'est sélectionnée, l'erreur 1014 est générée. Si le BLOB *paramImpression* ne contient pas des paramètres d'impression valides, l'erreur 1074 est générée.

Référence : [WR Param impression vers blob](#)