



# 4<sup>e</sup> Dimension 2004.3

## ADDENDUM

---

Bienvenue dans 4<sup>e</sup> Dimension version 2004.3. Ce document présente les nouveautés et modifications de cette version :

- **Module de sauvegarde** : le module de sauvegarde intégré de 4<sup>e</sup> Dimension propose plusieurs nouveautés (miroir logique, nouvelles commandes et nouvelles options).
- **Langage** : des modifications ont été apportées aux commandes FIXER PARAMETRE BASE et FIXER OPTION IMPRESSION, la commande MONTRER BARRES DEFILEMENT LISTBOX est renommée CHOIX VISIBLE BARRES DEFILEMENT.
- **4D Write** : 4D Write 2004.3 inclut des modifications relatives à la gestion des encadrements internes et au zoom.

## Module de sauvegarde

Dans 4<sup>e</sup> Dimension 2004.3, le module intégré de sauvegarde s'enrichit des fonctions et options suivantes :

- la possibilité de mettre en place une sauvegarde par miroir logique sur 4D Server à l'aide de deux nouvelles commandes,
- une nouvelle préférence de sauvegarde en cas d'échecs répétés de la sauvegarde,
- un nouveau mécanisme lorsque la taille du fichier d'historique atteint une taille critique,
- la nouvelle commande Fichier historique.

## Miroir logique

4D Server propose une solution intégrée permettant de mettre en place un système de sauvegarde des données par miroir logique. Cette solution est basée sur deux nouvelles commandes : Nouveau fichier historique et INTEGRER FICHER HISTORIQUE.

**Qu'est-ce qu'un miroir logique ?**

Le miroir logique est un mode de sauvegarde sophistiqué, principalement destiné aux bases de données critiques ou à forte charge d'exploitation.

Utiliser un miroir logique consiste à exploiter une base sur un premier poste, et à maintenir sur une deuxième machine une copie de cette base, périodiquement mise à jour. Les deux machines communiquent par le réseau, la machine en exploitation transmettant régulièrement à la machine miroir les évolutions de la base par l'intermédiaire du fichier d'historique.

De cette façon, en cas d'un incident sur la base en exploitation, il n'y a qu'à repartir de la base miroir pour reprendre très rapidement l'exploitation sans aucune perte de données. En outre, la base en exploitation n'est jamais "bloquée" par les opérations de sauvegarde.

**Pourquoi choisir la sauvegarde par miroir logique ?**

L'emploi d'un miroir logique correspond à des besoins spécifiques. La stratégie standard basée sur une sauvegarde périodique et le fichier d'historique constitue dans la plupart des cas une solution simple, fiable et peu coûteuse. La base est sauvegardée régulièrement (toutes les 24 heures en général). Durant la sauvegarde, la base reste accessible en mode lecture seulement. Cette période d'indisponibilité partielle est très courte, et même pour les bases de données importantes (plus de 2 Go) elle ne dépasse pas 5 minutes. Cette opération peut même être programmée en-dehors des périodes d'utilisation de la base.

Cependant, pour certains organismes, comme par exemple les hôpitaux, les bases de données critiques doivent être entièrement opérationnelles 24h/24. La base ne peut pas être en lecture seulement, même durant un laps de temps très court. Dans ce cas, la mise en place d'un miroir logique est une solution adaptée.

---

*Note* La base miroir ne reflète que les modifications apportées aux **données**. Ce mode de sauvegarde n'est donc pas adapté pour des bases en cours de développement, où les fréquentes modifications de structure rendront rapidement le miroir obsolète ou nécessiteront de multiples réactualisations de la structure de la base miroir.

---

**Principes de fonctionnement**

---

*Les fonctions de miroir logique ne sont disponibles qu'avec 4D Server.*

---

La mise en place d'un système de sauvegarde par miroir logique s'appuie sur deux nouvelles commandes : Nouveau fichier historique et INTEGRER FICHIER HISTORIQUE.

Les principes mis en oeuvre sont les suivants :

- La base est installée sur le poste 4D Server principal (poste en exploitation) et une copie identique de la base est installée sur le poste 4D Server miroir.
- Un test au démarrage de l'application (par exemple la présence d'un fichier spécifique dans le dossier 4D Extensions) permet de distinguer chaque version (en exploitation et en miroir) et donc d'exécuter les opérations appropriées.
- Sur le poste 4D Server en exploitation, le fichier d'historique est "segmenté" à intervalle régulier à l'aide de la commande Nouveau fichier historique. Aucune sauvegarde n'étant effectuée sur le serveur principal, la base est en permanence disponible en lecture-écriture.
- Chaque "segment" de fichier d'historique est envoyé sur le poste miroir, où il est intégré à la base miroir à l'aide de la commande INTEGRER FICHER HISTORIQUE.

La mise en place de ce système nécessite la programmation de code spécifique, notamment :

- un minuteur sur le serveur principal pour la gestion des cycles d'exécution de la commande Nouveau fichier historique,
- un système de transfert des "segments" de fichier d'historique entre le poste en exploitation et le poste miroir (utilisation de 4D Internet Commands pour un transfert via ftp ou messagerie, Web Services, 4D Open pour 4D...),
- un process sur le poste miroir destiné à superviser l'arrivée de nouveaux "segments" de fichier d'historique et à les intégrer via la commande INTEGRER FICHER HISTORIQUE,
- un système de communication et de gestion d'erreurs entre le serveur principal et le serveur miroir.

---

***La sauvegarde par miroir logique est incompatible avec les sauvegardes "standard" car l'emploi simultané de ces deux modes de sauvegarde entraîne la désynchronisation de la base en exploitation et de la base miroir. Par conséquent, vous devez veiller à ce qu'aucune sauvegarde, automatique ou manuelle, ne soit effectuée sur la base en exploitation ou sur la base miroir.***

---

## Nouveau fichier historique

Nouveau fichier historique → Texte

Paramètres	Type	Description
Résultat	Texte	← Chemin d'accès complet du fichier d'historique refermé

**Thème :** Sauvegarde

---

*Cette commande ne fonctionne qu'avec 4D Server. Elle ne peut être exécutée que via la commande Executer sur serveur ou dans une procédure stockée.*

---

La commande Nouveau fichier historique referme le fichier d'historique courant, le renomme et en crée un nouveau avec le même nom et au même emplacement que le précédent. Cette commande est destinée à la mise en place d'un système de sauvegarde par miroir logique (cf. [paragraphe "Miroir logique", page 1](#)).

La commande retourne le nom complet (chemin d'accès+nom) du fichier d'historique refermé (appelé "segment"). Ce fichier est stocké au même emplacement que le fichier d'historique courant (spécifié dans la page Configuration des préférences de l'application, thème Sauvegarde). La commande n'effectue aucun traitement (compression, segmentation) sur le fichier sauvegardé. Aucune boîte de dialogue n'apparaît.

Le fichier est renommé avec les numéros de sauvegarde courants de la base et du fichier d'historique, sur le modèle suivant :

*NomBase[NumSvge-NumSvgeHisto].4DL*. Par exemple :

- si la base MaBase.4DD a été sauvegardée 4 fois, le dernier fichier de sauvegarde se nomme MaBase[0004].4BK. Le nom du premier "segment" de fichier d'historique sera donc MaBase[0004-0000].4DL.
- si la base MaBase.4DD a été sauvegardée 3 fois et que le fichier d'historique a été sauvegardé 5 fois depuis, le nom de la 6<sup>e</sup> sauvegarde du fichier d'historique sera MaBase[0003-0005].4DL.

Avant d'effectuer l'opération, 4D Server vérifie qu'aucune opération critique (transaction ou indexation) n'est en cours. Si c'est le cas, 4D Server tient compte des paramétrages d'attente définis dans la page Sauvegarde des préférences de l'application, thème Sauvegarde.

En cas d'erreur, la commande génère un code que vous pouvez intercepter à l'aide de la commande APPELER SUR ERREUR.

**Référence :** INTEGRER FICHIER HISTORIQUE

## INTEGRER FICHIER HISTORIQUE

INTEGRER FICHIER HISTORIQUE (cheminAccès)

Paramètres	Type	Description
cheminAccès	Texte	→ Nom ou chemin d'accès du fichier d'historique à intégrer

**Thème :** Sauvegarde

---

*Cette commande ne fonctionne qu'avec 4D Server. Elle ne peut être exécutée que via la commande Executer sur serveur ou dans une procédure stockée.*

---

La commande INTEGRER FICHIER HISTORIQUE provoque l'intégration dans la base courante du fichier d'historique dont le nom ou le chemin d'accès complet a été passé dans le paramètre *cheminAccès*. A l'issue de l'intégration, le fichier intégré devient le nouveau fichier d'historique courant de la base. Cette commande est destinée à la mise en place d'un système de sauvegarde par miroir logique (cf. [paragraphe "Miroir logique"](#), page 1).

Seuls les fichiers d'historique non archivés (extension .4DL) peuvent être intégrés par cette commande. Aucune boîte de dialogue n'apparaît, toutefois une barre de progression est affichée à l'écran.

Vous pouvez passer dans *cheminAccès* un chemin d'accès absolu ou relatif au dossier de la base. Si vous passez une chaîne vide dans ce paramètre, une boîte de dialogue standard d'ouverture de fichier s'affiche, permettant de désigner le fichier à intégrer. Si la boîte de dialogue est annulée, aucun fichier n'est intégré et la variable système *OK* prend la valeur 0.



En cas d'erreur, la commande génère un code que vous pouvez intercepter à l'aide de la commande APPELER SUR ERREUR. A noter que si au moins un enregistrement est verrouillé, la commande ne fait rien et l'erreur 1420 est générée.

- Note* Lors de l'utilisation de cette commande, il est du ressort du développeur :
- d'installer la base miroir sur le poste miroir et de s'assurer que le fichier de données ne sera pas modifié autrement que par l'intégration d'un fichier d'historique via la commande INTEGRER FICHIER HISTORIQUE. Pour détecter qu'il s'agit de la version miroir de la base, il est possible de placer un fichier dans le dossier *4D Extensions* ou le dossier de la base et de tester sa présence par exemple au cours de la méthode base Sur ouverture. Si le fichier est présent, le mode miroir est activé.
  - de mettre en place un système de communication entre la base en exploitation et la base miroir afin d'organiser l'envoi et la réception des segments de fichier d'historique. Il est possible d'utiliser pour cela un Web service, le plug-in 4D Open for 4D ou 4D Internet Commands.
  - de gérer les éventuelles erreurs de transmission entre les deux bases.

**Référence :** [Nouveau fichier historique](#)

**Scénario d'exploitation d'un miroir logique**

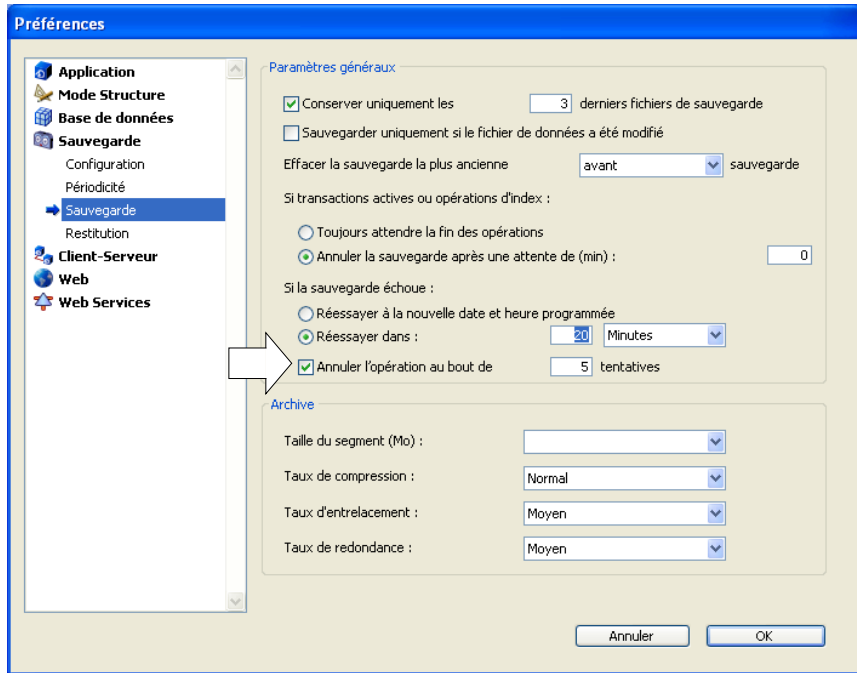
Le scénario suivant illustre, du point de vue de chaque poste 4D Server, la mise en place d'un système de sauvegarde avec miroir :

<b>Poste en exploitation</b> 	<b>Poste miroir</b> 
Démarrage de l'application, sauvegarde du fichier de données et activation (si nécessaire) du fichier d'historique. 4D crée le fichier <i>MaBase.4DL</i>	
On quitte l'application	
Copie de tous les fichiers de la base (fichier d'historique inclus) sur le poste miroir	
Démarrage de l'application et passage en exploitation.	Démarrage de l'application miroir. 4D Server demande le fichier d'historique : sélection du fichier <i>MaBase.4DL</i> transféré depuis le poste en exploitation.
Décision de mettre à jour le miroir (par exemple, après une certaine durée d'exploitation)	

Exécution de la méthode contenant la commande <b>Nouveau fichier historique</b> . Le fichier sauvegardé est nommé <i>MaBase[0001-0000].4DL</i>	
Envoi par programmation du fichier <i>MaBase[0001-0000].4DL</i> au poste miroir (utilisation de 4DIC, 4D Open pour 4D, etc.)	
La base est en exploitation	Détection qu'un fichier est en attente d'intégration. Exécution de la méthode contenant la commande <b>INTEGRER FICHIER HISTORIQUE</b> pour intégrer le fichier <i>MaBase[0001-0000].4DL</i> . Ce fichier devient le nouveau fichier d'historique courant.
Incident sur le poste, la base de données est inutilisable. Décision de passer sur la machine miroir	
Copie du fichier d'historique courant <i>MaBase.4DL</i> sur le poste miroir, dans le dossier de réception habituel	
Réparation de la machine...	Détection qu'un fichier est en attente d'intégration. Exécution de la méthode contenant la commande <b>INTEGRER FICHIER HISTORIQUE</b> pour intégrer le fichier <i>MaBase.4DL</i> . Ce fichier devient le nouveau fichier d'historique courant.
	La base est en exploitation
La machine est réparée	On quitte la base
Remplacement des fichiers de la base par ceux de la base miroir	
Démarrage de l'application. 4D Server demande le fichier d'historique : sélection du fichier <i>MaBase.4DL</i> transféré depuis le poste miroir.	Démarrage de l'application miroir.
La base est en exploitation	

## Nouvelle préférence de sauvegarde

Une nouvelle préférence permet de configurer le nombre maximal de tentatives infructueuses de sauvegarde : **Annuler l'opération au bout de N tentatives**. Cette option est accessible dans la page Sauvegarde/Sauvegarde de la boîte de dialogue des Préférences :



Ce paramètre est utilisable uniquement lorsque l'option **Réessayer dans...** est cochée. Il permet de définir le nombre de fois que le module de sauvegarde réessaiera de lancer la sauvegarde en cas d'échec.

Si, à l'issue du nombre d'essais défini, la sauvegarde n'a pas pu être effectuée, elle est annulée et l'erreur 1401 est générée ("Le nombre maximal de tentatives de sauvegarde est atteint, la sauvegarde automatique est temporairement désactivée"). Dans ce cas, aucune nouvelle sauvegarde automatique ne sera lancée tant que l'application n'aura pas été redémarrée ou qu'une sauvegarde manuelle n'aura été effectuée avec succès.

Ce paramètre est utile notamment pour éviter qu'en cas d'impossibilité prolongée de la sauvegarde (nécessitant une intervention humaine), l'application n'effectue inutilement de nombreuses tentatives au détriment de ses performances.

Par défaut, ce paramètre n'est pas coché.



## Sauvegarde automatique du fichier d'historique en cas de taille critique

Un nouveau mécanisme a été mis en place dans 4<sup>e</sup> Dimension 2004.3 : lorsque le fichier d'historique approche de la taille critique de 2 Go, 4D déclenche automatiquement une sauvegarde de la base, permettant de refermer l'historique courant et d'en ouvrir un nouveau. Ce mécanisme permet d'éviter que la taille de ce fichier n'altère les performances de l'application.

La sauvegarde effectuée dans ce cas utilise les paramètres de sauvegarde courants et est identique à une sauvegarde manuelle.

## Fichier historique

Fichier historique → Alpha

### Paramètres Type Description

Cette commande ne requiert pas de paramètre

Résultat Alpha ← Nom long du fichier d'historique de la base

La nouvelle commande Fichier historique retourne le nom long (c'est-à-dire le chemin d'accès complet au fichier, y compris son nom) du fichier d'historique courant de la base ouverte.

Si la base fonctionne sans fichier d'historique, la fonction retourne une chaîne vide et la variable système OK prend la valeur 0.

Si la base fonctionne avec un fichier d'historique, la variable système OK prend la valeur 1. Le chemin d'accès retourné par la commande est exprimé avec la syntaxe de la plate-forme courante.

---

*Note* Si vous exécutez cette commande depuis un poste 4D Client, seul le nom du fichier d'historique est retourné, pas le nom long.

---

**Thème** : Sauvegarde.

## Langage

En plus des commandes ajoutées pour la gestion d'un miroir logique (cf. [paragraphe "Miroir logique", page 1](#)) et la nouvelle commande Fichier historique (voir ci-dessus), diverses modifications relatives au langage ont été effectuées dans 4<sup>e</sup> Dimension 2004.3.

## Editeur de méthodes

Dans l'éditeur de méthodes, le nouveau raccourci **Alt+double-clic** (Windows) ou **Option+double-clic** (Mac OS) sur le nom d'une méthode projet appelée permet de l'ouvrir directement dans une nouvelle fenêtre.

## Taille de la pile des process

La gestion interne de la mémoire allouée à la pile des process a été modifiée dans 4<sup>e</sup> Dimension 2004.3. Il est désormais conseillé d'allouer au moins 64 000 octets à la pile de chaque process. Des valeurs inférieures peuvent conduire à l'apparition de l'erreur "Taille de la pile insuffisante pour terminer la méthode courante".

La taille de la pile des process est définie via les commandes Nouveau process et Executer sur serveur. Dans les versions précédentes de 4<sup>e</sup> Dimension, ce paramètre était géré de façon dynamique et, de fait, une taille supérieure à la taille demandée était généralement utilisée. Ce principe apportait plus de confort de développement mais, dans des conditions de fortes charges d'utilisation, pouvait dégrader de façon significative les performances des applications, notamment 4D Server. Désormais, la taille définie est strictement utilisée, ce qui peut nécessiter une mise à niveau de la valeur de la taille des piles.

## MONTRER BARRES DEFILEMENT LISTBOX

Sa portée ayant été étendue, la commande MONTRER BARRES DEFILEMENT LISTBOX a été renommée **CHOIX VISIBLE BARRES DEFILEMENT** et a été déplacée du thème "List Box" vers le thème "Propriétés des objets".

En effet, cette commande est utilisable avec plusieurs objets de formulaire pouvant comporter des barres de défilement :

- les list box,
- les zones de défilement,
- les sous-formulaires.

Les deux paramètres *horizontal* et *vertical* sont obligatoires. A noter que dans le cas des zones de défilement, le paramètre *horizontal* doit être passé, il sera simplement ignoré.

## CHOIX VISIBLE BARRES DEFILEMENT

CHOIX VISIBLE BARRES DEFILEMENT est le nouveau nom de la commande MONTRER BARRES DEFILEMENT LISTBOX (cf. paragraphe précédent).

**Thème** : Propriétés des objets.

## 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
Enreg événements debogage	34	Application 4D

### ■ Sélecteur = 34 (Enreg événements debogage)

Valeurs possibles : 0, 1 ou 2 (0 = ne pas enregistrer, 1 = enregistrer, 2 = enregistrer en mode détaillé)

Description : Démarrage ou arrêt de l'enregistrement séquentiel des événements générés au niveau programmation de 4D, destiné au débogage de l'application. Par défaut, la valeur est 0 (pas d'enregistrement des événements).

Diverses informations sont enregistrées, notamment :

- pour chaque événement, le nombre de millisecondes depuis la création du fichier et le numéro de process (*n*)
- l'exécution de chaque commande 4D (*cmd*) et chaque appel de plug-in (*plugInName*) ; le niveau de la pile est indiqué dans ce cas (*n*)
- chaque appel de méthode projet (*meth*), de méthode objet (*obj*) et de méthode formulaire (*form*).
- lorsque le mode détaillé est activé (*valeur* = 2), des informations supplémentaires relatives aux plug-ins sont enregistrées : événements dans les zones de plug-ins (*EventCode*) et appels de 4D par les plug-ins (*externCall*).

Les événements sont stockés dans un fichier nommé

"4DDebugLog.txt" automatiquement placé à côté du fichier de structure de la base. Chaque événement est systématiquement inscrit dans le fichier avant son exécution, ce qui garantit sa présence dans le fichier même lorsque l'application quitte inopinément. A noter que le fichier est effacé et réécrit à chaque lancement de l'application.

L'option peut être activée dans tout type d'application 4D

(4<sup>e</sup> Dimension monoposte, 4D Server, 4D Client, 4D Runtime), en interprété ou en compilé.

---

***Cette option est proposée uniquement à des fins de débogage, elle ne doit pas être utilisée en production car elle peut entraîner la dégradation des performances de l'application et la saturation du disque dur.***

---

Voici un exemple de contenu du fichier de débogage :

```
-- Startup on Friday, October 07, 2005 03:29:26 PM --
0 [1] Log level: 1
250 [1] form: Sortie; event: OnLoad
250 [1] end_form
7050 [1] obj: vRecNum; event: OnLoad
7083 [1]      (1) cmd: Table du formulaire courant.
7083 [1]      (1) cmd: Numero dans selection.
7083 [1]      (1) cmd: Table du formulaire courant.
7083 [1]      (1) cmd: Numero dans selection.
7083 [1]      (1) cmd: Chaîne.
7083 [1]      (1) cmd: Table du formulaire courant.
7083 [1]      (1) cmd: Enregistrements trouves.
7083 [1]      (1) cmd: Chaîne.
7083 [1] end_obj
7083 [1] form: Entrée; event: OnLoad
7083 [1] end_form
8900 [1] obj: Bouton1; event: OnClicked
8900 [1]      (1) plugInName: _4D_Pack; cmd: AP AVAILABLE MEMORY.
8900 [1] end_obj
10250 [1] obj: Bouton2; event: OnClicked
10250 [1]      (1) cmd: ALERTE.
11666 [1] end_obj
18983 [6] 4DActionUrl: /4DACTION/testWeb
18983 [6] onWebAuthenticationCall: /4DACTION/testWeb
18983 [6] end_onWebAuthenticationCall
18983 [6] meth: testWeb
18983 [6]      (1) cmd: ALERTE.
19000 [6] end_meth
19000 [6] end_4DActionUrl
19883 [6] 4DActionUrl: /4DACTION/testWeb
19883 [6] onWebAuthenticationCall: /4DACTION/testWeb
19883 [6] end_onWebAuthenticationCall
19883 [6] meth: testWeb
19883 [6]      (1) cmd: ALERTE.
19900 [6] end_meth
19900 [6] end_4DActionUrl
```

**Thème :** Définition structure.

## FIXER OPTION IMPRESSION, LIRE OPTION IMPRESSION

FIXER OPTION IMPRESSION(option; valeur1 {; valeur2})

LIRE OPTION IMPRESSION(option; valeur1 {; valeur2})

Une nouvelle constante d'option est disponible avec 4<sup>e</sup> Dimension 2004.3 :

option ( <u>Constante</u> )	valeur1	valeur2
Option masquer progression impr (14)	0 = Ne pas masquer (défaut), 1 = Masquer	-

Cette nouvelle option permet de masquer toutes les fenêtres système de progression d'impression pour le process courant. Passez 1 dans *valeur1* pour masquer les fenêtres de progression et 0 pour rétablir leur affichage (fonctionnement par défaut).

Cette option est particulièrement utile dans le cadre des impressions en PDF sous Mac OS X.

---

*Note* Il existe déjà une option d'affichage **Progression de l'impression** accessible via la boîte de dialogue des préférences (page Application/Options). Toutefois, elle est globale à l'application et ne masque pas toutes les fenêtres sous Mac OS X.

---

**Thème** : Impressions.

## 4D Write

Les modifications suivantes ont été effectuées dans 4D Write 2004.3 :

- nouvelle gestion des encadrements internes,
- conservation du zoom,
- nouveau bouton de suppression globale des tabulations.

## Compatibilité des documents

Les documents créés avec une version de 4D Write antérieure à la 2004.3 sont entièrement compatibles avec la nouvelle version du plug-in.

Toutefois, le format de stockage interne des documents ayant été modifié, les documents 4D Write créés ou convertis en version 2004.3 ne pourront pas être relus par une version précédente du plug-in.

## Gestion des encadrements

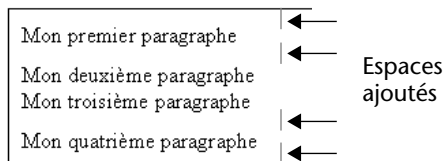
La gestion des encadrements internes des paragraphes a été modifiée dans 4D Write 2004.3. Cette modification implique de nouvelles icônes dans la barre d'outils "Encadrements" et de nouvelles constantes.

## Nouveau fonctionnement

Dans les versions précédentes de 4D Write, l'insertion d'encadrements entraînait l'ajout d'un espace au-dessus et en-dessous de chaque paragraphe concerné. Dans l'exemple suivant, l'encadrement "Contours" est attribué à une sélection de 4 paragraphes :

4D Write ancienne version

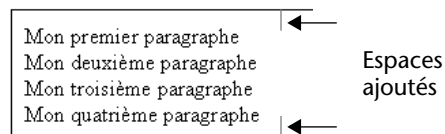
Mon premier paragraphe  
Mon deuxième paragraphe  
Mon troisième paragraphe  
Mon quatrième paragraphe





Ce fonctionnement pouvant s'avérer problématique dans certains cas, désormais 4D Write n'ajoute plus d'espace au-dessus et en-dessous du paragraphe encadré, mais uniquement du côté concerné :

4D Write 2004.3

Mon premier paragraphe  
Mon deuxième paragraphe  
Mon troisième paragraphe  
Mon quatrième paragraphe



Ce nouveau fonctionnement est également appliqué lors de l'utilisation des boutons **Encadrement haut** et **Encadrement bas** dans la barre d'outils "Encadrements".

Toutefois, à des fins de compatibilité, l'ancien mécanisme a été conservé. Il est désormais accessible via les nouveaux boutons **Encadrement intérieur haut**  et **Encadrement intérieur bas**  de la barre d'outils "Encadrements".

## Constantes "WR Propriétés de texte"

Le nouveau fonctionnement des encadrements se traduit par des modifications dans le thème de constantes "WR Propriétés de texte" :

- Les constantes existantes (43 et 44) ont été renommées respectivement wr encadrement inter haut et wr encadrement inter bas.

Ces constantes conservent le fonctionnement des versions précédentes de 4D Write (ajout d'espace au-dessus et en-dessous des paragraphes). Ce principe préserve la compatibilité des documents existants.

- Deux nouvelles constantes ont été ajoutées : wr encadrement haut (46) et wr encadrement bas (47).  
Ces constantes correspondent au nouveau fonctionnement des encadrements haut et bas dans 4D Write 2004.3.

### Constantes "WR Commandes"

Le nouveau fonctionnement des encadrements se traduit également par des modifications dans le thème de constantes "WR Commandes" :

- Les constantes existantes (1006 et 1008) ont été renommées respectivement wr cmd encadrement inter haut et wr cmd encadrement inter bas. Ces constantes correspondent aux nouveaux boutons de la barre d'outils "Encadrements" de 4D Write 2004.3. Elles conservent le fonctionnement des versions précédentes de 4D Write (ajout d'espace au-dessus et en-dessous des paragraphes). Ce principe préserve la compatibilité des applications existantes.
- Deux nouvelles constantes ont été ajoutées : wr cmd encadrement haut (1015) et wr cmd encadrement bas (1016).  
Ces constantes correspondent aux boutons d'encadrement standard de la barre d'outils "Encadrements". Le fonctionnement de ces boutons a été modifié dans 4D Write 2004.3.

### Appliquer le zoom à l'ouverture

Il est désormais possible d'ouvrir une zone 4D Write avec une valeur de zoom personnalisée, sauvegardée lors de la fermeture précédente de la zone. La valeur de zoom courante est toujours sauvegardée avec la zone, mais par défaut elle n'est pas utilisée à l'ouverture et la zone est ouverte avec un zoom de 100 %.

Pour activer l'application du zoom à l'ouverture, vous devez utiliser la nouvelle constante wr utiliser valeur zoom stockée (18) avec la commande WR FIXER PROPRIETE ZONE.

- Pour ouvrir la zone avec la valeur de zoom précédente (sauvegardée dans le document), exécutez l'instruction suivante :  
WR FIXER PROPRIETE ZONE (zone; wr utiliser valeur zoom stockée;1)
- Pour rétablir l'ouverture de la zone à 100 % (fonctionnement par défaut), exécutez l'instruction suivante :  
WR FIXER PROPRIETE ZONE (zone; wr utiliser valeur zoom stockée;0)

---

*Rappel* Passez 0 dans le paramètre *zone* pour appliquer la propriété à toutes les zones 4D Write ouvertes par la suite.

---

## **Suppression des tabulations**

Un nouveau bouton **Tout supprimer** a été ajouté dans la boîte de dialogue de gestion des tabulations. Il permet de supprimer toutes les tabulations du ou des paragraphe(s) sélectionné(s) dans le document.