



# Conversion de vos bases 4D en ***4D v11 SQL***

**Configuration matérielle et logicielle requise 5****Conversion en 4D v11 SQL de bases en ancienne version 5**

L'Assistant de conversion .....	6
Informations sur le fichier de structure : .....	6
Informations sur le fichier de données : .....	7
Informations sur les segments : .....	7
Informations sur les disques : .....	7
Option : .....	8
En cas de problèmes lors de la conversion .....	8
Ce qui se passe à la conversion : .....	9
Nouveaux fichiers .....	9
Nouveaux dossiers : .....	9
Utilisez le Centre de Sécurité et de Maintenance après conversion.....	9
Liaison « forte » entre fichier Structure et fichier de données : .....	10
Ce qui disparaît après conversion : .....	10
<b>Ne sont plus utilisés : 11</b>	
Le Mode utilisation .....	11
L'accès direct aux ressources de 4D et aux ressources système .....	11
Système de substitution : .....	12
4DK# - STR# - cicn - Raccourcis-clavier – Macros .....	12
Les sous-tables .....	12
Les anciens composants .....	13
Les segments de données .....	14
Les macros 4D 2003/2004 .....	14
La personnalisation de l'icône de la fenêtre d'identification via une ressource .....	15
L'assignation des Groupes et Utilisateurs pour les opérations sur les Tables .....	15
Le choix du mode pour les images sur fond .....	16
<b>Ne sont plus supportés : 16</b>	
<b>Les modifications de comportement 16</b>	
Formulaires Table et Formulaire Projet .....	16
CPU et petite image bitmap en mosaïque en fond sur un formulaire .....	16
Suppression possible des tables et des champs .....	16
Les transactions .....	17
Les transactions imbriquées .....	17
Les index .....	17
Nouvel opérateur de comparaison : « contient mot-clé » .....	18
Les ensembles .....	19
Conteneur de Glisser-déposer et Conteneur de données .....	19
Les menus .....	19
Particularité du menu Edition .....	20
Listes hiérarchiques .....	20
Les triggers .....	20
Les plug-ins .....	20
Plugin SDK .....	21
Structure virtuelle .....	21
Les images .....	21
Typage des variables image .....	21
Stockage d'informations dans un champ image : .....	22
<b>Les compatibilités dans les Préférences : 22</b>	
Les compatibilités Développement : .....	23
Les compatibilités Web .....	24
Plateforme .....	24
Nouvelles options de compatibilités : .....	24
Interdire de glisser des données ne provenant pas de 4D .....	24
Les transactions imbriquées .....	25
Mode unicode .....	25
Attention : A faire impérativement pour convertir en 4D v11 une base 4D 2004 qui contient des données dans différentes langues .....	26
Définir le point et la virgule comme caractère d'emplacement dans les formats numériques : .....	27
Options de compatibilités qui ne sont plus présentes : .....	28
<b>Les commandes 28</b>	

Les nouvelles commandes .....	28
Relatives à la prise en charge de l'Unicode .....	28
relatives au moteur SQL et serveur SQL.....	28
Relatives aux transactions imbriquées.....	29
relatives aux recherches et tris .....	29
relatives aux nouveaux formulaires projet.....	29
relatives au glisser-déposer .....	29
relatives au conteneur de données.....	29
relatives aux menus.....	29
Relatives aux listes hiérarchiques.....	30
Relatives aux ListBox .....	30
Relatives aux impressions.....	30
Relatives à l'environnement 4D : .....	30
Relatives à l'environnement système : .....	30
Relatives à l'interface utilisateur : .....	30
Relatives aux méthodes.....	30
Relatives aux chaînes de caractères, XLIFF et regex.....	30
Relatives à la définition de structure.....	30
Relatives au Web.....	31
Relatives au XML.....	31
relatives aux images.....	31
Relatives aux Outils : .....	31
Les commandes dont la syntaxe a été modifiée : .....	31
Relatives aux menus : .....	31
Relatives aux listes hiérarchiques : .....	31
Relatives aux ListBox.....	31
Relatives aux impressions.....	31
Relatives aux composants : .....	32
Relatives à l'environnement de 4D : .....	32
Relatives à l'environnement système : .....	32
Relatives à la saisie et l'interface utilisateur : .....	33
Relatives aux chaînes de caractères : .....	33
Relatives à la définition de structure : .....	34
Relatives au Web.....	34
Relatives au XML.....	34
Relatives aux Documents système.....	34
Relatives aux BIObs .....	34
Relatives aux ressources .....	34
Relatives aux communications : .....	35
Relatives aux Web services .....	35
Relatives au process .....	35
Les commandes dont le nom a été modifié : .....	35
Thèmes divers .....	35
Relatives au conteneur de données : .....	35
Relatives au Menu : .....	35
Relatives aux listes hiérarchiques.....	36
Relatives à la définition de structure.....	36
Relatives à l'Import/export.....	36
Relatives aux tableaux .....	36
Les commandes dont le comportement a été modifié : .....	36
Commandes pour lesquelles le premier paramètre, la table, est maintenant obligatoire : .....	38
Les commandes qui changent de thèmes .....	38
Les commandes optimisées .....	38
Les commandes obsolètes .....	39
Les commandes dont l'utilisation est devenue inutile .....	39
Les syntaxes obsolètes.....	39
Les commandes sur les sous-enregistrements .....	39
<b>Les intégrations (partielles) 39</b>	
4D Insider remplacé par le « Glisser-déposer système » et par le « Chercher-remplacer » : .....	40
4D Tools remplacé par le CSM .....	41
4D Pack .....	41
Commandes 4D Pack obsolètes.....	41

Commandes toujours présentes mais qu'il est conseillé de ne plus utiliser :.....	42
Commande améliorée .....	43
Commandes identiques.....	43
<b>Les Raccourcis clavier</b>	<b>45</b>
Développement .....	45
Enregistrement.....	45
<b>Annexe 1 : Les codes langues:</b>	<b>47</b>

## Configuration matérielle et logicielle requise

- **Mac** : IMac Minimum / MacOS X 10.4.5.  
512 Mo de RAM minimum – 1 Go recommandé  
Version certifiée : 10.4.10

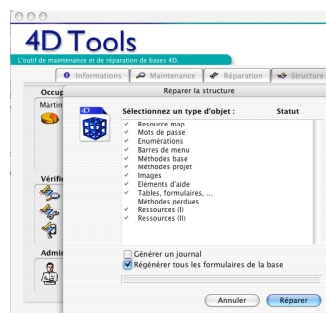
- **Windows** : Pentium II minimum / Windows XP  
512 Mo de RAM minimum – 1 Go recommandé

Version certifiée : Windows Serveur 2003 R2, Windows XP SP2, Windows VISTA

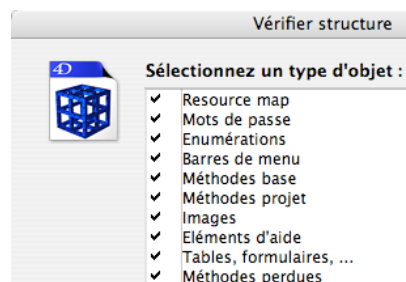
## Conversion en 4D v11 SQL de bases en ancienne version

Vous pouvez convertir les bases version 6.5 (voire 6.0) , 2003, ou 2004 directement en 4D v11 SQL.

Il est pourtant conseillé, au préalable, d'utiliser 4D Tools dans la version d'origine pour faire une vérification de la structure et des données. Profitez-en, si vous ne l'avez pas déjà fait, pour régénérer les formulaires avec l'option « **Régénérer tous les formulaires de la base** ». Cette option n'est accessible que dans la version 4D 2004 :



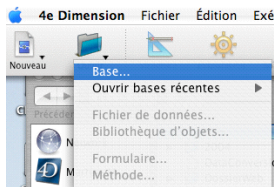
Si votre base vient d'une très ancienne version, faites une vérification de la structure avec l'option « Méthodes perdues cochée » :



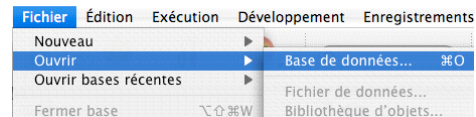
Lancez 4D v11 SQL :



Apparaît alors la barre de menus de 4D. Vous pouvez ouvrir une base avec l'outil **Ouvrir**:

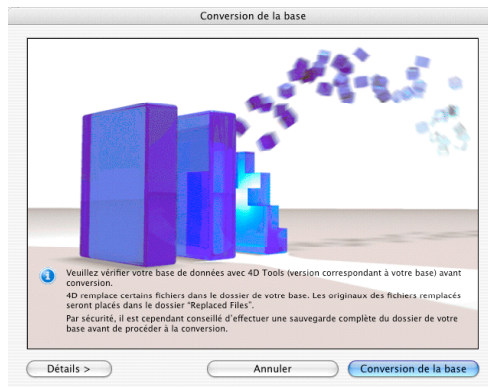


ou en passant par le menu :



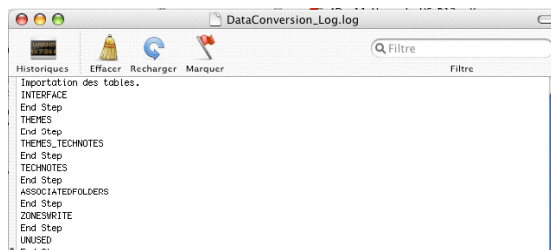
et choisissez le .4DB à convertir. L'assistant de conversion apparaît.

## L'Assistant de conversion



Un assistant de conversion vous est proposé. Il effectue une **copie** de la base d'origine (structure, données, fichier d'historique et fichier d'historique vierge) dans un dossier nommé "**Replaced files (Conversion)**".

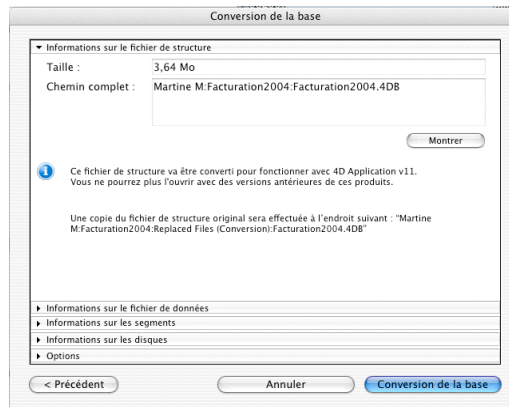
Les différentes étapes de la conversion sont consignées dans un fichier "**DataConversion\_Log.log**" qui est placé à côté du fichier de structure converti.



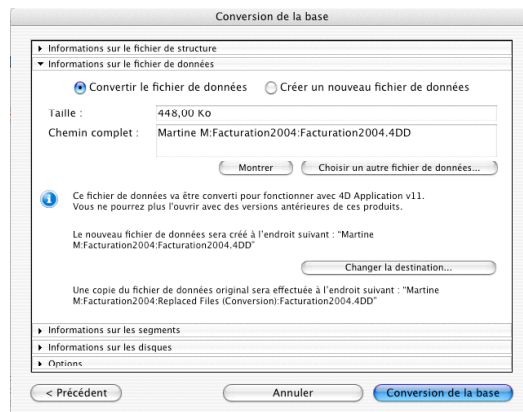
Ce fichier peut renvoyer les erreurs suivantes :

1012 : si vous avez converti une base avec plus d'un niveau de sous-tables

### Informations sur le fichier de structure :

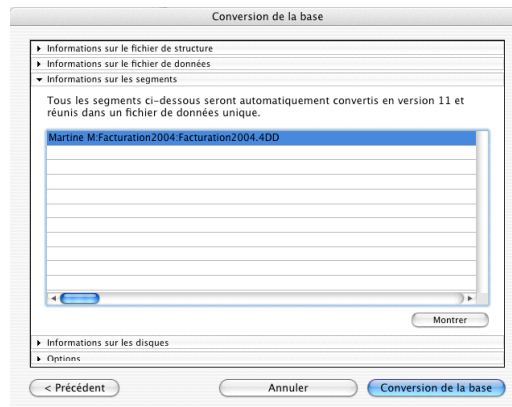


**Informations sur le fichier de données :**



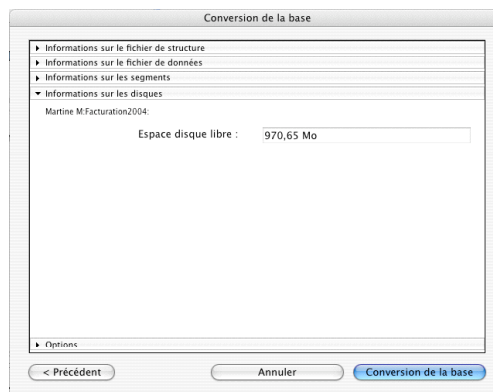
Vous pouvez créer un nouveau fichier de données

**Informations sur les segments :**



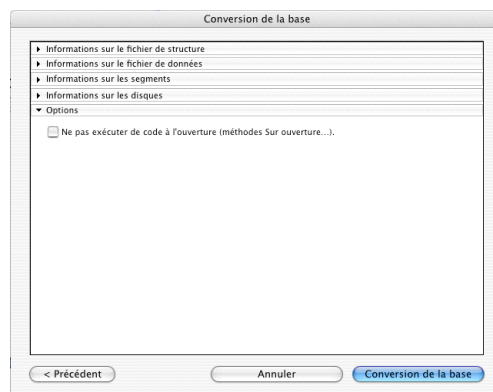
Les segments de données n'existent plus en 4D v11 SQL. A la conversion, tous les segments sont regroupés. La conversion ne pourra pas se faire s'il manque un segment.

**Informations sur les disques :**



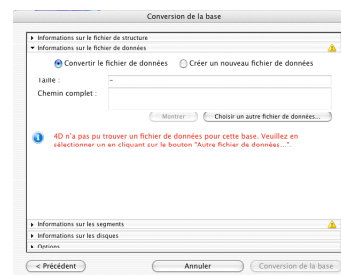
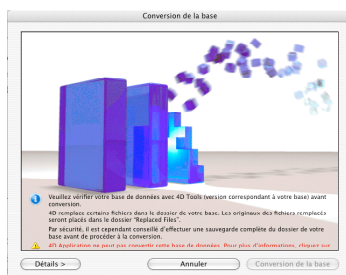
**Option :**

C'est ici que vous pourrez choisir de ne pas exécuter le code à l'ouverture :



C'est une option importante car elle permet de ne pas passer dans la méthode base "Sur ouverture" au premier lancement de la base convertie.

En cas de problèmes rencontrés, des messages d'erreurs vous indiqueront la marche à suivre, comme par exemple dans les copies d'écran ci-dessous qui concerne une erreur de fichier de données non-trouvé :



**En cas de problèmes lors de la conversion**

- générez un nouveau fichier de données ;
- sélectionnez : "Ne pas exécuter le code à l'ouverture de la base" dans les options de conversion ;
- Ouvrez le Centre de Sécurité et de Maintenance : vérifiez les enregistrements et les index, puis vérifiez l'application.
- installez les plug-ins si nécessaire (Dossier Plugins et Win4DX/Mac4DX)
- vérifiez en interprété et en compilé

**Note :**

Le mode interprété en 4D v11 SQL est moins tolérant que celui de la version 4D 2004.



## Ce qui se passe à la conversion :

### Nouveaux fichiers

Après conversion en version 4D v11 SQL, les bases de données 4D contiennent plusieurs fichiers supplémentaires :

- NomBase.4DIndy : contient l'index de la structure.
- NomBase.4DIndx (facultatif) : contient les index des données.
- DataConversion\_Log.log: ce fichier stocke les différentes étapes de la conversion et les éventuelles anomalies rencontrées. Si ce fichier mentionne des problèmes, utilisez le CSM pour réparer la base.

### Note :

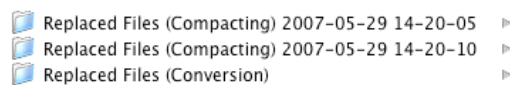
Les bases créées en 4D v11 SQL sont automatiquement placées dans un dossier suffixé **.4dbase**.  
Sous MacOS, les bases sont ainsi directement présentées sous forme de package.

Pour obtenir un package d'une base convertie, il suffit de d'ajouter le suffixe .4dbase à un dossier et de placer tous les fichiers nécessaires dans ce dossier.

### Nouveaux dossiers :

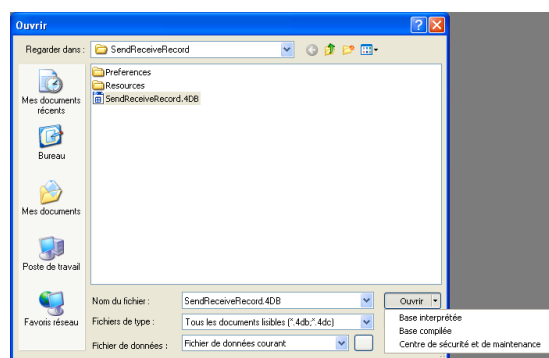
- Nouveaux Dossiers : Replaced files :

Différentes manipulations peuvent créer des dossiers nommés **Replaced files** : ces dossiers contiennent la copie des fichiers originaux de la base avant conversion ou avant compactage, ou avant réparation.



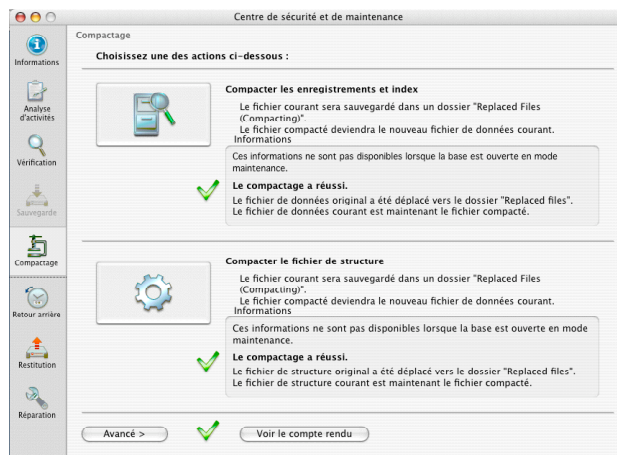
- Nouveau dossier : Macros v2 (voir Macros)
- Nouveau dossier : Resources (voir Ressources)

### Utilisez le Centre de Sécurité et de Maintenance après conversion

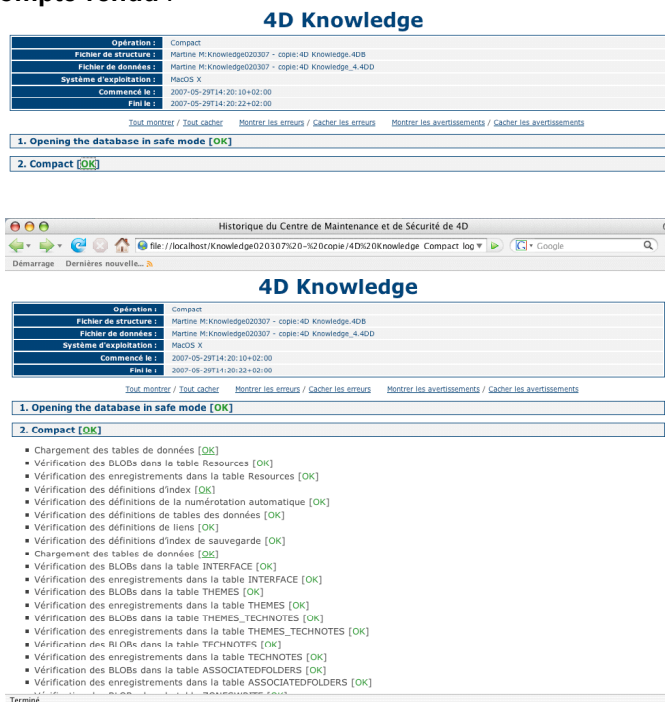


Le Centre de Sécurité et de Maintenance remplace désormais 4D Tools. Il faut lancer 4D Developer pour y accéder :





Vous pouvez ici réparer, compacter la structure et les enregistrements. Voici un exemple de compte-rendu si vous cliquez sur **Voir le compte-rendu** :



## Liaison « forte » entre fichier Structure et fichier de données :

Lors de la conversion ou de la création d'une base en 4D v11 SQL, chaque base se voit attribuée un identifiant qui sert de lien entre structure et données. Donc une même base convertie deux fois de suite de 4D 2004 en 4D v11 n'aura pas deux fois le même identifiant : le deuxième .4DD ne pourra pas être ouvert avec le premier .4DB et inversement : un message signale que le fichier de données ne correspond pas à la structure.

La ressource WEDD peut toujours être mise en place dans les préférences pour relier une structure et un fichier de données.

## Ce qui disparaît après conversion :

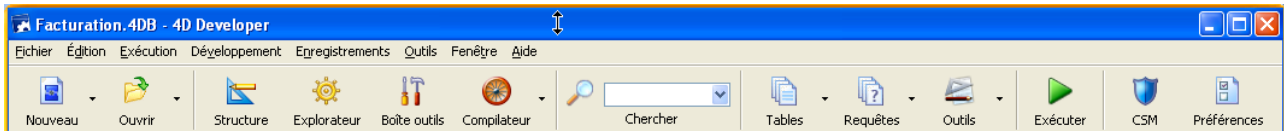
- les paramétrages du Mode trace
- « Publier la base au démarrage » sera décochée dans les Préférences Web, même s'il était coché

- Un lien récursif sur lui-même, plus précisément une relation entre un champ et lui-même, sera supprimé
- Les niveaux de sous-tables au-delà de 1

## Ne sont plus utilisés :

### Le Mode utilisation

Le Mode Structure et le Mode Utilisation ont été unifiés pour donner accès à la fois au Mode Développement et aux Enregistrements :



### L'accès direct aux ressources de 4D et aux ressources système

Conformément aux orientations définies par Apple et mises en oeuvre dans les versions de Mac OS les plus récentes, le concept de ressources est désormais obsolète et est progressivement abandonné. Certains développeurs ont pu tirer parti de ressources internes de 4<sup>ème</sup> Dimension pour leurs interfaces (par exemple les ressources contenant les noms des mois ou ceux des commandes du langage). Cette pratique, non officielle, est désormais proscrite. Dans la plupart des cas, il est possible d'utiliser d'autres moyens que les ressources internes de 4D (constantes, commandes du langage...).

Il est désormais interdit (et impossible) de faire appel aux ressources du système ou de l'application 4<sup>ème</sup> Dimension, que ce soit via les commandes ou via la syntaxe <:15000,5>.

Le code suivant est désormais obsolète :

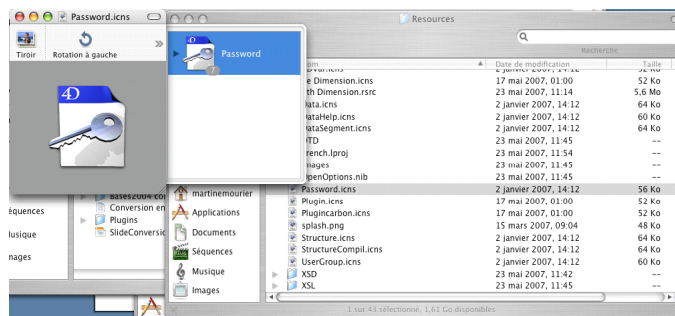
```
$Structure :=fichier structure
$Ressources :=Ouvrir fichier ressources ($Structure)
```

en 4D v11 SQL la commande **Ouvrir fichier ressources** renvoie toujours 0, car les ressources sont toujours ouvertes par 4D.

Un mécanisme de substitution a été mis en place pour limiter l'impact de cette modification sur le fonctionnement des bases de données tirant parti de ces ressources.

Pour maintenir la compatibilité, les anciens mécanismes continueront à fonctionner, avec quelques adaptations :

- s'ils sont présents, les fichiers de ressources sont toujours supportés par 4D et le principe de la chaîne des fichiers de ressources (l'ordre des ouvertures successives) restent valide. Cette chaîne inclut les fichiers .rsr et .4dr et les fichiers personnels ouverts avec les commandes 4D. Mais il n'est plus possible d'accéder aux ressources de 4D. Cette pratique, déjà déconseillée auparavant, est maintenant strictement interdite.
- Pour limiter l'impact de cette modification sur les bases existantes, un système de substitution a été implémenté, basé sur l'externalisation des ressources les plus fréquemment utilisées.



Il est conseillé de les dupliquer, si vous souhaitez les utiliser.

### Système de substitution :

- Placez tous les fichiers nécessaires dans un dossier nommé « **Resources** », que vous devez créer à côté de la structure.

Il est cependant fortement conseillé de faire évoluer les bases de données converties et de supprimer les appels aux ressources internes de 4<sup>ème</sup> Dimension.

**Note :**

Les bases créées en 4D v11 SQL n'ont plus de .RSR et de .4DR.

### 4DK# - STR# - cicc - Raccourcis-clavier – Macros...

Pour récupérer vos ressources, un composant “4Dpop Migration” sera disponible gratuitement au téléchargement. Il vous permettra de récupérer :

- vos **STR#** et les convertira en fichier **.XLIFF**
- vos constantes **4DK#** prises dans le fichier **.rsr** qui seront converties en plug-in **User\_Contants.bundle**
- vos raccourcis-claviers, pour les éditer et les modifier : fichier **4DShortcuts.xml** dans le dossier “Extensions”
- vos macros version 2004 pour les passer en UTF-8 (cross-plateforme) et convertir les caractères désormais interdits (**// ----- <>** etc....)
- etc....

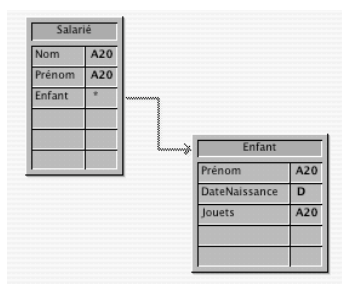
*Pour plus d'informations sur ce composant, nous consulter*

### Les sous-tables

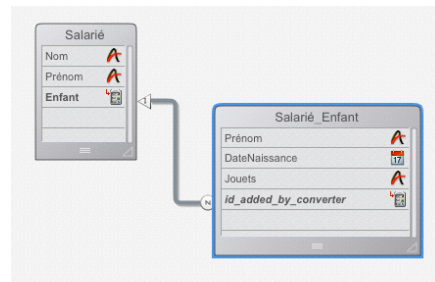
Il n'est plus possible de créer des sous-tables dans la version 4D v11 SQL.

Lors de conversion d'une base en v11, les sous-tables **premier niveau** sont transformées en tables standards avec un lien spécial. Attention, les niveaux suivants seront purement et simplement tronqués.

Par exemple, l'ancienne structure :



Devient :



La sous-table devient Table N, et la table d'origine devient Table 1

Le nom de la table nouvellement créée sera : NomTableParente\_NomSousTable et sera tronqué à 31 caractères !

Une clef d'appel est rajoutée dans la Table N, nommée : id\_added\_by\_convertir

Nom du lien aller : NomSousTable\_id\_added\_by\_convertir

Nom du lien retour : id\_added\_by\_convertir\_subtable\_1\_Enfant

L'ordre des tables en v11 n'est plus le même qu'en 2004, les sous-tables s'ajoutent aux tables existantes.

Si vous aviez des sous-tables dans votre base, **ATTENTION** aux points suivants :

- le nombre de tables après migration est supérieur en 4D v11 SQL.
- Si vous convertissez une base avec des sous-tables de niveau supérieur à 1, celles-ci ne seront pas récupérées, et seront tous simplement tronqué à un niveau.  
Dans ce cas l'erreur **1012** apparaîtra dans le log '**DataConversion\_Log.log**' ainsi d'ailleurs que dans la fenêtre de résultat d'erreurs à la fin de la conversion .
- tant que vous ne supprimez pas le lien spécial créé lors de la conversion, les commandes sur les sous-enregistrements continueront à fonctionner. Mais la suppression du lien spécial est irréversible : vous devrez retracer un lien classique et il faudra alors remplacer les commandes « sous-enregistrement » par les commandes sur les tables.

Il faudra donc prévoir une réécriture du code si vous utilisiez les numéros de tables ou si vous supprimez le lien.

Attention aussi à certaines commandes dont le comportement changent : **DUPLIQUER ENREGISTREMENT** ne duplique plus les sous-enregistrements, **ENVOYER ENREGISTREMENT** n'envoie plus des sous-tables et **RECEVOIR ENREGISTREMENT** ne les reçoit plus.

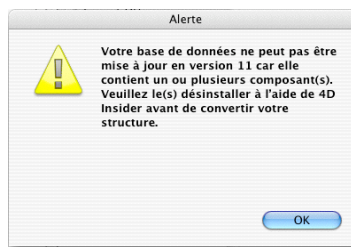
Par contre, désormais, les commandes **Champ** et **Nom du champ** retourne bien le numéro et le nom du sous-champ lorsque l'on dépointe le pointeur (et non plus ceux de la table principale).

## Les anciens composants

L'architecture des composants a été modifiée en profondeur. Un composant est désormais un simple fichier de structure, compilée ou non, sous forme de package (dossier suffixé .4dbase). Il suffit de le placer dans un dossier « Components » à côté de la structure.

Il n'est plus possible d'utiliser des tables dans un composant 4D v11 SQL (un base créée en 4D v11 SQL ne contient pas de table 1, et vous pouvez désormais utiliser des formulaires projet (non liés à une table).

Si un composant ancienne version est installé dans votre base, un message d'alerte vous avertit que la conversion est impossible :

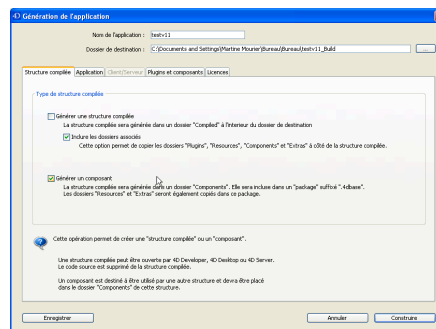


Les anciens composants doivent être désinstallés avec 4D Insider de la version de départ avant conversion.

### Note :

Dans un composant 4D v11 SQL on ne peut pas utiliser les tables et champs, les formulaires table et leurs méthodes formulaire associées, les formulaires utilisateurs, les méthodes base et triggers.

Les composants sont désormais des bases 4D : .4db, .4dc ou .4dbase. Pour les applications professionnelles, le .4dbase est préconisé. Vous pouvez l'obtenir à partir du dialogue de génération d'applications :

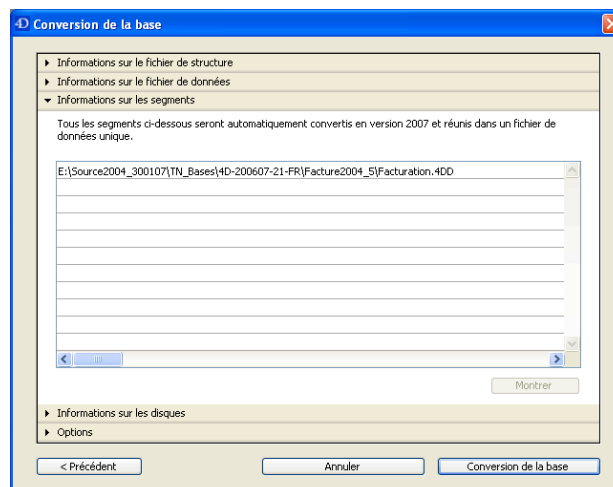


## Les segments de données

La taille du fichier de données est virtuellement illimitée (exceptées les limites imposées par le système). Il n'est plus possible de créer et d'utiliser plusieurs segments de données. Les anciens segments sont regroupés à la conversion.

Vérifiez que vous disposez de suffisamment de place sur le disque pour cela.

Si un segment est manquant, l'assistant vous demandera de le localiser manuellement. S'il ne peut pas être localisé, la conversion ne pourra pas se faire.



## Les macros 4D 2003/2004

4D v11 SQL charge les macros d'un nouveau dossier « Macros v2 » (dans le dossier actif de 4D, à côté de la structure, ou dans le dossier « component » pour les macros d'un composant).

les macros utilisées avec les versions 2003 et 2004 de 4<sup>ème</sup> Dimension ne sont pas compatibles avec 4<sup>ème</sup> Dimension v11. Les macros v11 doivent être entièrement conformes au standard XML. Il faut donc que les mentions :

```
< ?xml version= « 1.0 »... ?>
<http://www.4d.com/dtd/v11/Macros.dtd>
```

soient présentes.

La dtd se trouve :

Sous MacOS : 4D Developer :app :contents :dtd :macros.dtd

Sous Windows : 4D Developer/Resources/DTD/macros.dtd

Ne sont plus compatibles :

// pour les commentaires

----- les tirets

< ou > qui doivent être encodés : &lt ; et &gt ;

la balise <macros> est obligatoire

....

Un **mécanisme de conversion automatique** est mis en place. 4D v11 SQL copie automatiquement les macros 4D 2003 ou 4D 2004 dans le dossier « Macros v2 » . Les lignes de déclarations sont rajoutées au document. Mais des ajustements ponctuels peuvent être à envisager pour coller complètement avec le standard XML.

Nouvelles commandes :

**LIRE PARAMETRE MACRO**  
**FIXER PARAMETRE MACRO**

## La personnalisation de l'icône de la fenêtre d'identification via une ressource

Cette fonctionnalité ne fonctionne plus. Elle est remplacée par la simple copie d'un fichier .png  
Il suffit de nommer votre image « LoginImage.png » et de la glisser dans le dossier « Resources » à côté de la structure.

## L'assignation des Groupes et Utilisateurs pour les opérations sur les Tables

Pour les opérations sur les données comme Chargement, Ajout, Sauvegarde, Suppression, on ne peut plus assigner de groupe. De même, il n'y a plus de groupe propriétaire d'une Table. Ces contrôles deviennent inefficaces en cas de base convertie. Désormais les contrôles se font à différents niveaux (formulaires, menus, méthodes...) ou via des options spécifiques, par exemple le SQL (dans les Préférences )

Accès au serveur SQL

Lecture seulement (données)	<Tout le monde>
Lecture/écriture (données)	<Tout le monde>
Complet (données et structure)	<Personne>

NOTE  
Ces paramètres ne sont pris en compte que lorsque le système de mots de passe 4D est activé (le Super\_Utilisateur a un mot de passe)

**Note :**

Attention aux trous de sécurité par 4D Open.

## Le choix du mode pour les images sur fond

Il n'est plus possible de choisir le mode d'interaction entre les couleurs d'une image (format Image sur fond) et du fond via la fenêtre « Choix du mode ». Ce comportement qui avait été mis en place pour les images monochromes n'est plus adapté pour des images qui sont gérées désormais en mode natif.

## Ne sont plus supportés :

- passer un élément de tableau 2D à une commande qui demande une variable : renvoie désormais une erreur : ex. **Type**(MonTableau2D{1}{1})
- les faux tableaux dans les popup menus MacOS 7 : si la variable du popup était en réel et s'appelait par exemple « pop », on dessinait le contenu de la variable « pop »+**chaîne**(pop)
- l'opérateur paragraphe § qui permettait d'adresser une variable par son nom (une sorte de pointeur)  
exemple : a := § («vtoto »)  
Cet ancien opérateur était déjà rejeté par le compilateur.  
Exemple :  
ValeurDeA :=§("A")  
  
doit désormais s'écrire :  
ValeurDeA :=**Pointeur vers**("A")->
- **EFFACER VARIABLE** n'accepte plus la syntaxe permettant de donner un nom de variable entre guillemets. Fonctionne désormais à l'identique en interprété et en compilé.
- Les faux tableaux pour lesquels tab{« 42 »} ou tab{42} étaient équivalents à tab42
- Onglet associé à un tableau « toto », et un tableau booléen « toto\_ » pour activer/désactiver chaque élément de l'onglet.
- DDETools : Ce produit n'est plus supporté. Les alternatives sont :
  - VBA et **LANCER PROCESS EXTERNE** sous Windows
  - Applescript et **LANCER PROCESS EXTERNE** sous MacOS.

## Les modifications de comportement

### Formulaires Table et Formulaire Projet

Il existe désormais un type de formulaire qui ne dépend d'aucune table : le Formulaire Projet. Ce type de formulaire peut être utilisé à la fois comme Formulaire Sortie et comme Formulaire Entrée, contrairement à ce que laisse croire le pop-up de choix qui ne propose que le Formulaire Détaillé et Formulaire Impression Détaillé (ou Sans).

### CPU et petite image bitmap en mosaïque en fond sur un formulaire

En raison de la modification de la gestion des images, vous pouvez constater une forte augmentation du CPU utilisé en 4Dv11 en utilisant un formulaire avec petite image bitmap dupliquée en mosaïque en fond d'un formulaire (plus l'image est petite, plus elle doit être dupliquée, plus l'utilisation CPU est importante).

La solution est d'utiliser une image de plus grande taille (128 x 128 mini) qui prendra un peu plus de mémoire mais économisera le temps CPU.

### Suppression possible des tables et des champs



On peut désormais supprimer dans 4D v11 SQL les tables et les champs.  
Les formulaires d'une table supprimée sont transformés en Formulaires projet et se retrouvent dans la corbeille de l'explorateur.

Les numéros des tables et des champs supprimés ne sont ré-attribués qu'à la création.

<b>Commandes obsolètes :</b>	<b>Remplacées par :</b>
<b>Nombre de tables</b>	<b>Lire numero derniere table</b>
<b>Nombre de champs</b>	<b>Lire numero dernier champ</b>

**Commandes ajoutées :**  
**Est un numero de table valide**  
**Est un numero de champ valide**

## Les transactions

**VALIDER TRANSACTION** et **ANNULER TRANSACTION** ne touchent plus à la sélection courante.

Les versions précédentes réduisaient la sélection à zéro si au moins un enregistrement avait été créé durant la transaction.

Désormais, les enregistrements créés durant une transaction ne reçoivent plus de numéro temporaire (débutant à 18 000 000) mais un numéro standard, correspondant à la numérotation en cours dans la table. Ce numéro devient définitif en cas de validation de la transaction et est libéré en cas d'annulation (les enregistrements créés sont alors supprimés).

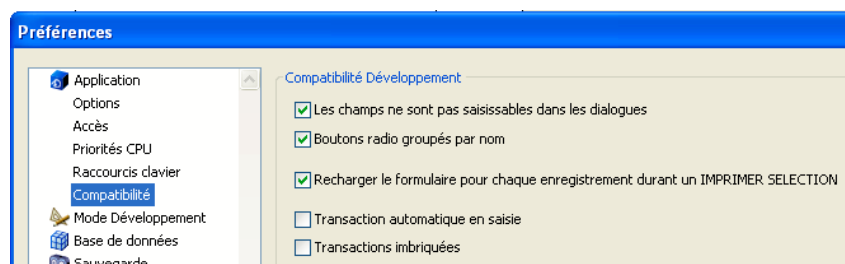
Les commandes fonctionnant avec les numéros d'enregistrements comme **SCAN INDEX**, **JOINTURE**, etc, peuvent désormais être utilisées dans les transactions.

## Les transactions imbriquées

Une préférence permet d'activer le mode multi-transactions dans 4D.

Avec 4D 2004, vous ne pouviez pas imbriquer des transactions dans d'autres. Si vous débutiez une transaction dans une autre, 4<sup>ème</sup> Dimension ignorait la seconde.

Cette option n'apparaît que pour les bases de données converties. Par défaut, elle n'est pas cochée.

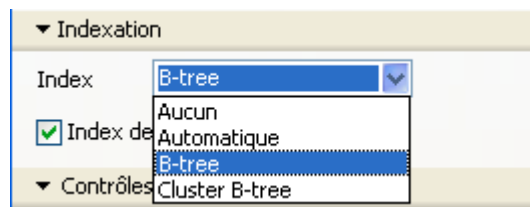


Cette option n'a pas d'effet sur les transactions effectuées dans le moteur SQL de 4D v11 SQL, qui lui est toujours multi-transactionnel.

## Les index

Il n'est plus obligatoire d'indexer les champs utilisés par un lien.

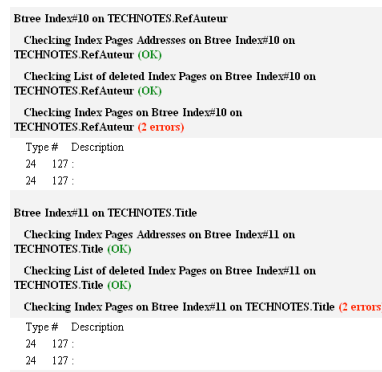
Par défaut vos index sont convertis en Beta-tree.



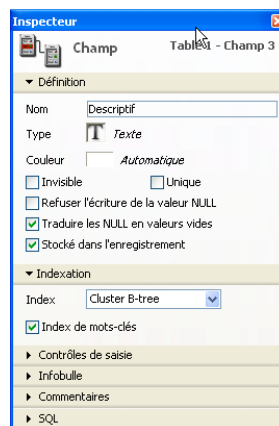
Les champs ajoutés au ancienne sous-table transformée en Table sont en Cluster B-Tree.

- **B-tree standard** : indexage polyvalent. Utilisé dans les versions précédentes de 4D.
- **Cluster B-tree** : plus efficace lorsque les mêmes valeurs reviennent souvent dans les données (par exemple : un champ booléen, ou un champ Genre qui contiendra Monsieur, Madame, ou Mademoiselle)
- **Automatique** (valeur par défaut en création) : 4<sup>ème</sup> Dimension sélectionne l'architecture en fonction du type de champ.

Voici un exemple de vérification des index avec le Centre de Maintenance :



- Index de **mots-cles** :



Ce nouveau type d'index est disponible pour les champs Alpha et Texte. Ils peut être combiné avec un index standard. 4D utilisera le plus approprié en fonction du contexte.

## Nouvel opérateur de comparaison : « contient mot-clé »

Pour ce nouveau type d'index, il existe un nouvel opérateur : % qui correspond à « Contient mot-clé »

Contient le mot « facile »

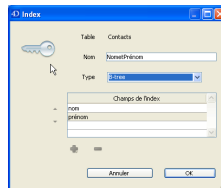
**CHERCHER**([Produits];[Produits]Description%"facile")

Contient le mot qui commence par « facile » :

**CHERCHER**([Produits];[Produits]Description%"facile@")

- **Index composites** :

Les index composites sont des index composés de plusieurs champs (deux ou plus). Ils ne peuvent être créés qu'à partir de la liste des index (bandeau des outils de structure) :



## Les ensembles

Lorsqu'un ensemble n'est plus à jour parce qu'il contient la référence à un enregistrement qui a été supprimé et que l'on utilise la commande **UTILISER ENSEMBLE**, 4D v11 SQL renvoie l'erreur suivante : -10503 : "Numéro d'enregistrement non valide"

Avec 4D 2004, ce n'était pas le cas : aucun numéro d'erreur n'est renvoyé.

**Appartient a ensemble** : prenait le pointeur de sélection (parfois pointant sur un autre enregistrement avec **EMPILER ENREGISTREMENT** par exemple) mais pas la fiche courante si elle était chargée, contrairement à **ADJOINDRE ELEMENT** – désormais le comportement est identique.

Pour information, **NOMMER ENSEMBLE** garde toujours le pointeur de sélection sans s'occuper de la fiche chargée.

## Conteneur de Glisser-déposer et Conteneur de données

Ces commandes peuvent être utilisées pour gérer le conteneur (anciennement presse-papiers) et le glisser-déposer. 4D v11 SQL exploite en fait deux conteneurs : les données coupées (ou copiées) et les données en cours de glisser-déposer.

Le conteneur de Glisser-déposer n'est accessible que dans le contexte des événements formulaire Sur debut glisser, Sur glisser ou Sur déposer, ou dans la Méthode base Sur déposer.

Le thème « Presse papiers » a été renommé « Conteneur de données »

La constante Données absentes dans presse papiers a été renommées Données absentes conteneur.

## Les menus

Désormais dans 4D v11 SQL, les menus sont hiérarchiques. Vous avez la possibilité de créer à la volée des menus et et barres de menus sans qu'ils existent en mode structure. Les menus et barres sont traités de la même façon.

Le nombre maximum de caractères passent de 31 à 150.

Le paramètre *LigneMenu*=-1, permet de désigner la dernière ligne ajoutée

### **Anciens noms (4D 2004.x)**

**CHANGER BARRE**  
**CACHER BARRE DE MENUS**  
**AFFICHER BARRE DE MENUS**  
**APPELER SUR A PROPOS**

### **Nouveaux noms (4D v11SQL)**

**FIXER BARRE MENUS**  
**CACHER BARRE MENUS**  
**AFFICHER BARRE MENUS**

*Transférée dans le thème "Interface utilisateur"*

**Titre menu**  
**Texte ligne menu**  
**CHANGER TEXTE LIGNE MENU**  
**Style ligne menu**  
**CHANGER STYLE LIGNE MENU**  
**Marque ligne menu**  
**MARQUER LIGNE MENU**  
**Raccourci clavier**  
**CHANGER RACCOURCI CLAVIER**

**Lire titre menu**  
**Lire texte ligne menu**  
**FIXER TEXTE LIGNE MENU**  
**Lire style ligne menu**  
**FIXER STYLE LIGNE MENU**  
**Lire marque ligne menu**  
**FIXER MARQUE LIGNE MENU**  
**Lire raccourci ligne menu**  
**FIXER RACCOURCI LIGNE MENU**

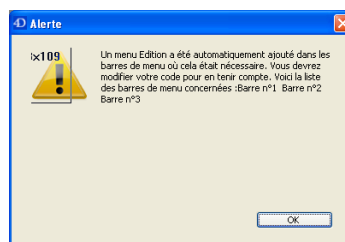
Pas de changement :

**Menu choisi**  
**Nombre de menus**  
**Nombre de lignes de menus**  
**INACTIVER LIGNE MENU**  
**ACTIVER LIGNE MENU**  
**AJOUTER LIGNE MENU**  
**INSERER LIGNE MENU**  
**SUPPRIMER LIGNE MENU**

## Particularité du menu Edition

L'option v 6.8 n'est plus disponible pour les menus dans 4D v11 SQL. Cette option permettait, dans les bases de données converties, de maintenir un ancien fonctionnement basé sur l'ajout automatique d'un menu **Edition** géré par le système.

Lors de la conversion d'une base de données dans laquelle l'option v 6.8 était cochée, le menu **Edition** est ajouté automatiquement et une alerte vous avertit de cela :



**Attention** : il vous faudra alors modifier la valeur du paramètre *menu* dans les instructions de gestion et d'exécution des menus. En effet, avec l'ancien fonctionnement, le menu **Edition** n'était pas comptabilisé, en 4D v11 SQL, il est comptabilisé.

## Listes hiérarchiques

Modification structurelle. Plus de souplesse.

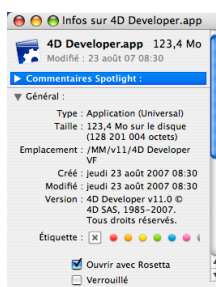
## Les triggers

Petit changement de comportement sur le trigger **Sur sauvegarde nouvel enregistrement**. En 4D 2004 lorsque l'on appelait la commande **Numero enregistrement** dans ce trigger, celle-ci retournait **-3**. En 4D v11 SQL, la commande retourne le numéro d'enregistrement qui sera affecté à la fiche nouvellement créée.

## Les plug-ins

Les plug-ins version 2004 sont compatibles (4D ou tierces parties) avec 4D v11 SQL.

MacIntel : Tous les plug-ins 4D v11 SQL sont **natifs MacIntel**, **exceptés 4D Draw et 4D for OCI** qui doivent être utilisés sous Rosetta. Pour cela, choisissez Lire les informations sur l'application 4D Developer.app et cochez « ouvrir avec Rosetta ».



## Plug-in SDK

EX\_FIND\_PROCID et EX\_CALL\_BY\_PROCID travaillaient avec les index dans le tableau des méthodes. Ces points d'entrée travaillent maintenant avec des ID internes, positifs ou négatifs, mais non nuls.

## Structure virtuelle

Une structure virtuelle utilise les noms des tables et champs mis en place avec **FIXER TITRES TABLES** et **FIXER TITRES CHAMPS**. Ces noms sont utilisés par les dialogues standards de 4D (recherche, tri, état rapides, etc). Le comportement a été modifié en 4D v11 SQL : désormais la structure virtuelle n'est utilisée que lorsque ces dialogues sont appelés avec les commandes du langage. Les plug-ins, aussi, accèdent toujours à la structure virtuelle :

	Structure	Table ou champs invisibles
Mode développement et environnement enregistrement	Structure réelle	visibles
Plug-ins	<b>Structure virtuelle</b>	<b>invisibles</b>
Mode application	<b>Structure virtuelle</b>	<b>invisibles</b>

## Les images

Les images sont désormais stockées au format natif.

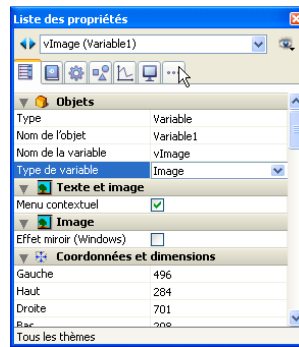
L'ancienne commande **LISTE TYPES IMAGES** est conservée pour des raisons de compatibilité. Toutefois, elle nécessite la présence de QuickTime et ne donne pas accès aux formats gérés en natif par 4D. Son intérêt est limité et elle est avantageusement remplacée par **LISTE CODECS IMAGES**.

## Typage des variables image

Si, en mode interprété 4D v11 SQL, vous constatez que vos variables images ne sont pas correctement affichées, assurez-vous qu'elles soient correctement typées :

Pour cela, vous disposez de deux solutions :

- vous pouvez la typer au niveau du formulaire via le le popup menu de la Liste des propriétés :



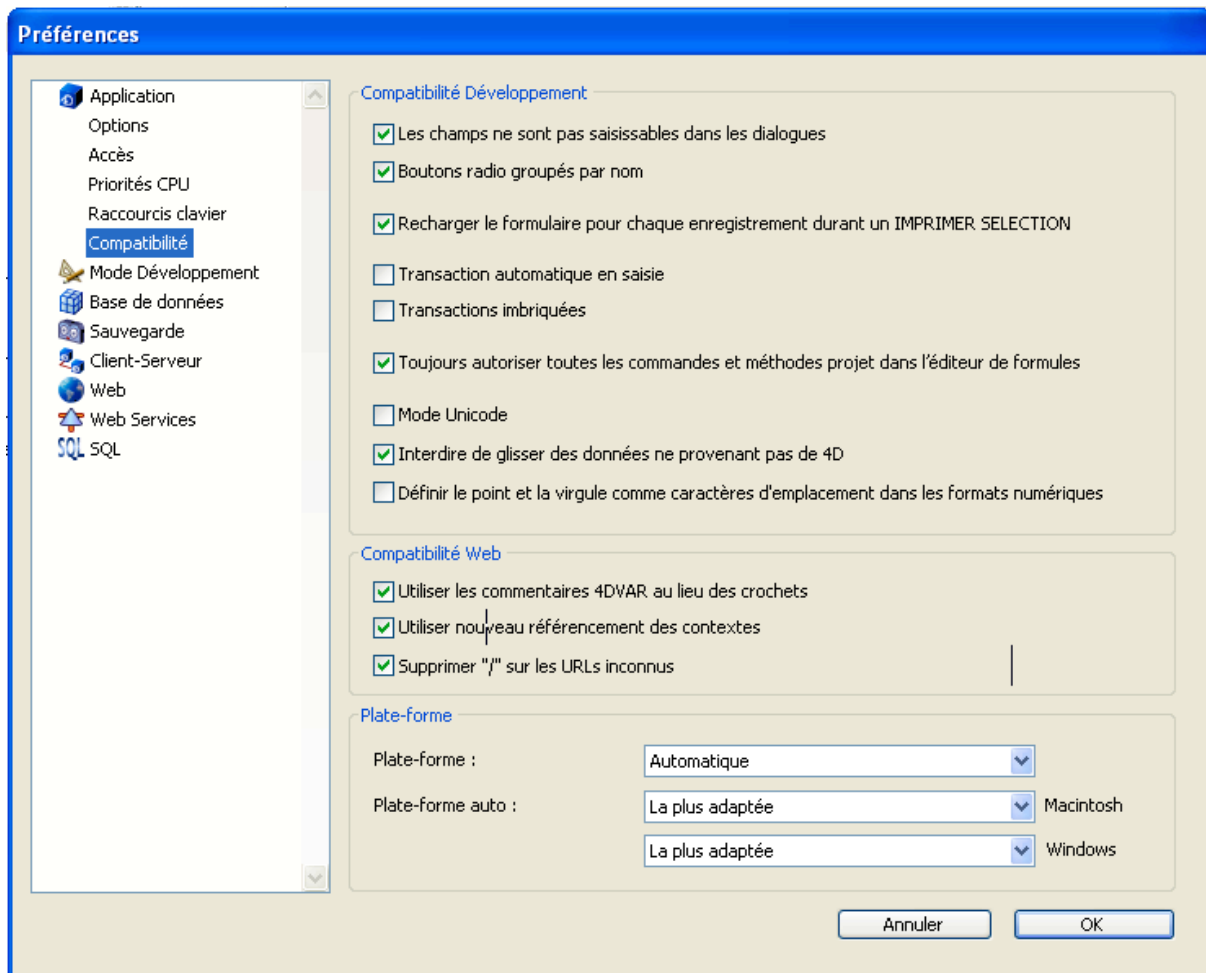
- Ou utiliser une directive de compilation : **C\_IMAGE**(vImage) qui doit être exécutée **avant** le chargement du formulaire c'est-à-dire avant même l'évènement formulaire Sur chargement.

Les versions précédentes de 4D étaient plus "tolérantes" quant à l'initialisation des variables en général, et des variables images en particulier dont l'affichage dans les formulaires est régi par de nouveaux mécanismes natifs.

### Stockage d'informations dans un champ image :

A la fin d'une image, 4D ajoutait les infos X/Y et mode d'affichage, qui n'ont plus lieu d'être maintenant. Ces 6 octets sont supprimés pendant la conversion. Si pour des besoins légitimes, vous aviez stocké des blobs dans les champs Image, sachez qu'ils perdent une partie de leurs informations à la conversion.

## Les compatibilités dans les Préférences :



## Les compatibilités Développement :

Options de compatibilités déjà présentes en 4D 2004 :

- Les champs ne sont pas saisissables dans les dialogues (cochée dans les **bases converties** et décochée pour les bases créées)
- Boutons radio groupés par nom : il n'y a plus d'obligation à faire commencer la variable bouton-radio par la même chaîne pour les associer. Il suffit désormais de simplement les grouper. Cette option n'existe que dans les **bases converties**. Elle est cochée par défaut. Les bases créées en 2004 utilisent le nouveau fonctionnement.
- Recharger le formulaire pour chaque enregistrement durant un **IMPRIMER SELECTION**.

Cette option n'apparaît que pour les anciennes **bases de données converties**. Elle est cochée par défaut.

Ce principe permettait de réinitialiser automatiquement tous les paramètres des objets que le développeur avait pu modifier par le langage dans l'événement formulaire **Sur impression corps**. Afin d'optimiser les performances, ce mécanisme a été supprimé dans 4<sup>ème</sup> Dimension 2004. Le développeur 4D doit désormais réinitialiser lui-même les paramètres qu'il souhaite dans la méthode formulaire — ce fonctionnement est identique à celui des formulaires listes avec l'événement **Sur affichage corps**. Vous pouvez toutefois conserver l'ancien mécanisme à l'aide de l'option « Recharger le formulaire pour chaque enregistrement durant un **IMPRIMER SELECTION** ». Les bases de données créées en 4D v11 utilisent le nouveau mode.

- "Transactions automatique en saisie" : Cette option n'apparaît que pour les anciennes **bases de données converties** en version 2004. Elle est décochée par défaut. Elle permettait de débiter

automatiquement une transaction quand un formulaire de saisie était ouvert et possédait un formulaire inclus. Option conçue pour les utilisateurs de 4D First qui passaient en 4D. Cette option était déjà déconseillée en 2003.

- « Toujours autoriser toutes les commandes et méthodes projets dans l'éditeur de formules » : Dans l'éditeur de formule, seules sont désormais affichés quelques thèmes des commandes 4D et aucune de vos méthodes projet. Sauf si vous utilisez la commande **FIXER METHODES AUTORISEES** (voir documentation) ou si vous cochez cette option.

## Les compatibilités Web

- Utiliser les commentaires 4DVAR au lieu des crochets : dans les anciennes versions du serveur Web, on employait la syntaxe suivante : [MaVar], solution propriétaire à remplacer par <!--4DVAR MaVar-->
- Utiliser nouveau référencement des contextes : évite l'envoi de n° de contexte par le serveur web de 4D au navigateur pour chaque élément d'une page.
- Supprimer "/" sur les URLs inconnus : ex. <http://127.0.0.1/Contacts/Ajouter> retournait dans \$1 : « Contacts/Ajouter » dans les versions précédentes. En 4D 2004 tous les URLs débutent par "/" (sauf en cochant cette option...)
- Connexions persistantes : si une page HTML comportait 10 images, cela entraînait auparavant 10+1 connexions en http 1.0. Cette option permet de maintenir une seule connexion.

## Plateforme

Dans le dialogue des préférences, cette zone n'apparaît que dans les bases de données **converties**. Dans les précédentes versions de 4<sup>ème</sup> Dimension, il était possible de "forcer" cette apparence au niveau de la base de données, des formulaires et/ou des objets — chaque niveau héritant par défaut du précédent. Ce mécanisme est conservé pour des raisons de compatibilité.

Dans les bases de données créées avec 4<sup>ème</sup> Dimension version 2004 et suivantes, l'interface est automatiquement choisie par 4<sup>ème</sup> Dimension en fonction de la plate-forme courante. Les formulaires et objets utilisés sont dessinés de manière native. L'interface peut être paramétrée uniquement au niveau des formulaires ou des objets et seules deux options sont disponibles : **Système** et **Impression** pour les formulaires, **Lié au formulaire** pour les objets.

Dans les bases de données converties depuis des versions précédentes de 4<sup>ème</sup> Dimension, des options supplémentaires sont disponibles (Automatique, Mac OS 7, Windows 3.11, NT 3.51, Windows 95/98/2000, NT 4, Windows 95/98/2000, NT 4, Mac OS 9 et Thème Mac). Ces options correspondent à l'ancien mode de prise en charge de l'interface de plate-forme dans 4<sup>ème</sup> Dimension et il est désormais déconseillé de l'utiliser. Il en est de même avec les commandes

**Lire interface** et **FIXER INTERFACE**, toutes deux conservées pour des questions de compatibilité qui fonctionnent dans les bases converties uniquement. Ces commandes sont ignorées dans les bases créées en 4D 2004, l'interface de plate-forme étant gérée automatiquement.

Dans les **Préférences** des bases converties, Il est fortement conseillé de sélectionner l'option **Système** afin de mettre en oeuvre ce fonctionnement. Les deux autres pop-up menus sont alors désactivés, le nouveau mécanisme est actif. A noter qu'il vous sera peut-être nécessaire de modifier la propriété "Plate-forme" de vos formulaires pour en tirer parti, en choisissant par exemple l'option **Lié à la base**.

## Nouvelles options de compatibilités :

### **Interdire de glisser des données ne provenant pas de 4D**

dans la page **Application/Compatibilité** :



4D v11 SQL permet le glisser-déposer de sélections, d'objets ou de fichiers extérieurs à 4D, comme par exemple des fichiers image. Cette possibilité doit être prise en charge par le code de la base. Dans les bases de données converties depuis une version précédente de 4D, cette possibilité peut entraîner des dysfonctionnements si le code existant n'est pas adapté. Pour cette raison, une nouvelle option des Préférences permet d'interdire le déposer d'objets externes dans la base : **Interdire de glisser des données ne provenant pas de 4D**. Cochée par défaut dans les bases converties.

## Les transactions imbriquées

4D v11 SQL accepte désormais les transactions imbriquées sur un nombre de niveaux illimité. Ce nouveau fonctionnement est désactivé par défaut dans les bases converties (les transactions restent limitées à un seul niveau).

Si vous souhaitez bénéficier des transactions sur plusieurs niveaux dans une base convertie, vous devez l'indiquer explicitement en cochant l'option.

Pour infos : Les transactions SQL sont toujours multi-niveaux.

## Mode unicode

Cette option est cochée par défaut dans une nouvelle base 4D v11 SQL, non cochée dans une base convertie.

4D v11 SQL utilise l'Unicode pour :

- le moteur de la base de données et le serveur SQL
- les variables
- le web
- de nombreux objets de structure : menus, liste, structure, utilisateur et groupes, tips.... (mais pas pour les formulaires – utilisez le XLIFF pour la localisation – ni pour les méthodes).

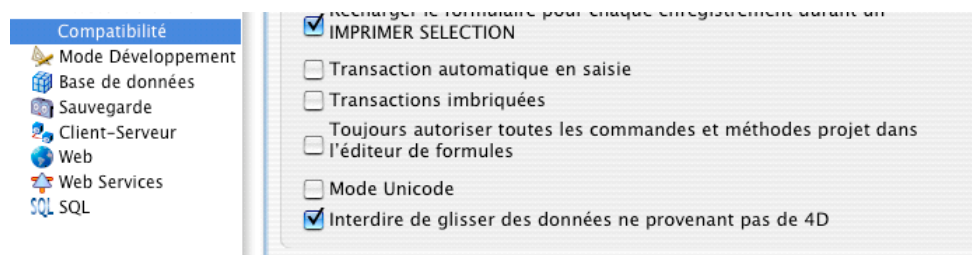
Préférences :

Le support des caractères Unicode a été étendu dans 4D v11. De nombreuses commandes ont été réécrites.

Désormais le moteur de la base de données, le langage et de nombreux éléments d'interface (menus, listes, etc) peuvent stocker et manipuler les caractères Unicode.

Lorsque le mode Unicode est activé, la propriété "Configuration du clavier" des objets de formulaire est ignorée.

Pour des raisons de compatibilité, 4D v11 SQL peut encore fonctionner avec l'ancien mode, basé sur un jeu de caractères ASCII Mac, avec une nouvelle option dans les préférences :



4D v11 SQL utilise maintenant de l'UTF-16 (et s'aligne ainsi sur les OS Windows et Mac) et parfois l'UTF-8, notamment pour le Web et les imports/exports, qui peut présenter des avantages (meilleure lisibilité pour les caractères courants - a-Z, 0-9 – et plus compact). En Unicode, les valeurs de 1 à 127 correspondent très exactement aux caractères ASCII.

Cette prise en charge de l'Unicode entraîne différentes modifications : changement de paramètre ou nouveaux paramètres pour certaines commandes, voire nouvelles commandes :

	<b>Non Unicode</b>	<b>Unicode – UTF 16</b>
Champ Alpha	Champ Alpha	Champ Alpha
Champ Texte	Champ Texte	Champ Texte
<b>C_TEXTE</b>	Type Texte	Type texte
<b>C_ALPHA</b>	Type variable chaîne	Type Texte Paramètre de longueur de chaîne ignoré
<b>TABLEAU TEXTE</b>	Type tableau texte	Type tableau texte
<b>TABLEAU ALPHA</b>	Type tableau chaîne	Type tableau texte Paramètre de longueur de chaîne ignoré
<b>Caractere</b>	<b>Caractere</b> (Code Ascii) 0-255	<b>Caractere</b> (valeur UTF-16) 0-65535
<b>Code ascii</b> devient <b>Code de caractere</b>	<b>Code de caractere</b> Renvoie un code ASCII	<b>Code de caractere</b> Renvoie un caractère Unicode
<b>Longueur, Chaîne, Position, Supprimer chaîne, sous chaîne, remplacer caractères</b>		Travaillent sur les caractères UTF-16
Syntaxe [[1]]		Retourne le énième caractère UTF-16 et non plus le énième octet
<b>Mac vers Windows, Windows vers Mac, Mac vers ISO, ISO vers Mac</b>		Ne fonctionnent plus

Nouvelles commandes : <b>CONVERTIR DEPUIS TEXTE</b> <b>Convertir vers texte</b>	-	<b>CONVERTIR DEPUIS TEXTE</b> (Texte4D ;jeu caracteres ;blob converti) <b>Convertir vers texte</b> (blob ;jeu caractère)-> texte
Nouveaux noms de constantes et nouvelles constantes : <b>BLOB vers texte</b> <b>TEXTE VERS BLOB</b>	<u>Chaîne en C</u> <u>Chaîne pascal</u> <u>Texte avec longueur</u> <u>Texte sans longueur</u>	<u>Mac Chaîne en C</u> <u>Mac Chaîne pascal</u> <u>Mac Texte avec longueur</u> <u>Mac Texte sans longueur</u> <u>UTF8 Chaîne en C</u> <u>UTF8 Texte avec longueur</u> <u>UTF8 Texte sans longueur</u>

Ces nouveaux mécanismes n'entraînent pas de modification particulière pour les langues dont les caractères sont codés sur un seul octet (langues occidentales) mais simplifiera les traitements dans les langues à caractères codés sur deux octets (telles que le japonais).

## Attention : A faire impérativement pour convertir en 4D v11 une base 4D 2004 qui contient des données dans différentes langues

Attention, si vous utilisez du multilingue (notamment des langues stockées sur 2 octets comme le chinois, Japonais, etc) dans vos champs, voici la démarche à suivre :

Durant la conversion 4D 2004 -> 4D v11 SQL, 4D convertit les données en Unicode. De façon à ne pas perdre d'informations, il est important de connaître le jeu de caractères d'origine. Ceci ne peut pas être fait totalement de façon automatique. Le développeur devra spécifier le jeu de caractère. Cette information est donnée via un fichier texte placé à côté du fichier de données à convertir « **multilang.txt** ». L'encodage doit être ANSI ou Mac Roman (ni UTF8 ou Unicode)

Description de ce fichier :

N°Table ; N° champ ; N° sous-champ (optionnel) ; code langue

Chaque ligne doit se terminer par un retour chariot (CR ou CRLF). Les lignes vides et espaces sont autorisés.

Ce fichier doit se trouver à côté du fichier de données AU MOMENT DE LA CONVERSION.

**Exemple :**

Si le développeur veut forcer les langages suivants :

**Greek** en Table3-champ3 et Table4-sous champ7–champ1,  
**Russian** en Table3-champ4 et Table4-sous champ7–champ2,

Le fichier « **multilang.txt** » contiendra les informations suivantes :

3;3;1049

3;4;1032

4;7;1;1049

4;7;2;1032

Les codes des langues sont donnés dans le fichier “**keyboardmapping.xml**” (voir annexe 1 du présent document).

**Base convertie en v11 sans fichier « multilang.txt » :**


English	Chinese	Russian	Japanese	German
Test	тiX § e	ипедлнг	ВзГТВ@lBcB=Br	DYrch
Another One	Ъ Ъ J @» § T §	гауффуичьдб	В†В†В†B§BcBcBcB®	AllCs

**La même base convertie en v11 avec le fichier “Multilang.txt” à côté du fichier de données durant la conversion :**



English	Chinese	Russian	Japanese	German
Test	差舞又	ипедлнг	さサおんでゆ	Dürch
Another One	哈搭器三卅	гауффуичьдб	ああえういいいお	Allès

Avec cette solution, la conversion fonctionne pour tous les langages présents.

**Définir le point et la virgule comme caractère d’emplacement dans les formats numériques :**

4D substitue automatiquement les caractères “.” et “,” dans les formats d’affichage des numériques par, respectivement, le séparateur des milliers et le séparateur décimal définis dans le système d’exploitation.

Dans les versions précédentes de 4D, le format “###,##0.00” était un format valide pour un système américain, mais lorsqu’il était appliqué à une valeur numérique affichée sur un système français ou suisse, le résultat était incorrect.

Désormais, en cochant cette nouvelle option des préférences et en utilisant systématiquement le point et la virgule comme séparateurs dans les formats numériques, 4D remplacera ces caractères par leur équivalent défini dans le système.

La commande **Num** accepte désormais un nouveau paramètre permettant de désigner un séparateur décimal spécifique. En outre, la nouvelle commande **LIRE FORMATAGE SYSTEME** permet de connaître la valeur courante des paramètres système régionaux.

**Attention :**

Dans les bases nouvellement créée en 4D v11 SQL, pour que votre base soit multi-OS, le séparateur des milliers à utiliser est la « , » et le séparateur des décimales à utiliser dans les formats est le « . », soit par exemple « ###,##0.00 € ». Vous aurez ainsi un formatage opérationnel sous n'importe quel OS.

Si vous souhaitez continuer à travailler avec des formatages classiques comme « ### ##0,00 € », décochez l'option dans les préférences, sinon modifiez vos formatages en « ###,##0.00 € »,

**Options de compatibilités qui ne sont plus présentes :**

- Utilisez la méthode Debut de la v5.x.x : désormais le code devant être exécuté à l'ouverture doit être placé dans la **Méthode base** Sur ouverture et/ou Sur démarrage serveur
- Utiliser les formules-fichiers de la v5.x.x : à remplacer par les **Triggers**
- Compatibilité 6.8 pour le rendu du texte (utilisé pour les anciens styles relief ou ombré, incompatible avec Quartz) – Si vous constatez des problèmes de lissage de polices, vérifiez que vous n'utilisez pas encore des polices bitmap, qui sont obsolètes.
- Ancien mécanisme du menu Edition (option v 6.8)  
L'option v 6.8 n'est plus disponible pour les menus dans 4D v11 SQL. Cette option permettait, dans les bases de données converties, de maintenir un ancien fonctionnement, basé sur l'ajout automatique d'un menu Edition géré par le système.  
Désormais, le menu Edition doit être entièrement géré comme les autres menus.  
Il sera automatiquement ajouté dans les barres où il n'existe pas (voir informations concernant le Menu Edition dans ce même document).

## Les commandes

**Les nouvelles commandes**

De nombreuses nouvelles commandes ont été rajoutées pour gérer de nouvelles fonctionnalités :

**Relatives à la prise en charge de l'Unicode**

**Lire traduction chaine**  
**Trouver regex**  
**CONVERTIR DEPUIS TEXTE**  
**Convertir vers texte**

**relatives au moteur SQL et serveur SQL**

**LISTE SOURCES DONNEES**  
**UTILISER BASE EXTERNE**  
**UTILISER BASE INTERNE**  
**Lire source donnees courante**  
**CHERCHER PAR SQL**  
**Est un cham Null**  
**FIXER CHAMP NULL**  
**LIRE DERNIERE ERREUR SQL**  
**LANCER SERVEUR SQL**  
**ARRETER SERVEUR SQL**

## Relatives aux transactions imbriquées

### Niveau de la transaction

## relatives aux recherches et tris

pour tirer parti du nouveau moteur et nouveau type de recherche et d'index

**CREER INDEX**

**SUPPRIMER INDEX**

**FIXER RECHERCHE ET VERROUILLAGE**

**Trouver dans champ**

*Recherche par mot-clés*

## relatives aux nouveaux formulaires projet

### PAS DE TABLE PAR DEFAULT

## relatives au glisser-déposer

4D permet le glissé-déposé de sélections, d'objets ou de fichiers extérieurs.

Nouvel événement formulaire Sur debut glisser

Nouvelle Méthode base : Sur déposer

## relatives au conteneur de données

Nouvelles signatures 4D :

**FIXER FICHER DANS CONTENEUR**

**Lire fichier dans conteneur**

**LIRE TYPE DONNEES DANS CONTENEUR**

Comm.4d.text.native

Com.4d.text.utf16

Com.4d.text.rtf

Com.4d.picture.pict

Com.4d.picture.png

Com.4d.file.url

Etc..

Les types sur 4 caractères (TEXT, PICT, etc. sont conservés par compatibilité.

## relatives aux menus

**Creer menu**

**EFFACER MENU**

**LIRE LIGNES MENU**

**Lire reference barre menu**

**LIRE ICONE LIGNE MENU**

**FIXER ICONE LIGNE MENU**

**Lire methode ligne menu**

**FIXER METHODE LIGNE MENU**

**LIRE PROPRIETE LIGNE MENU**

**FIXER PROPRIETE LIGNE MENU**

**Lire modificateurs ligne menu**

**Pop up menu dynamique**

**FIXER REFERENCE LIGNE MENU**

**Lire reference ligne menu**

**Lire reference ligne menu selectionnee**

## Relatives aux listes hiérarchiques

**FIXER POLICE ELEMENT**  
Lire police element  
Chercher dans liste  
**FIXER ICONE ELEMENT**  
**LIRE ICONE ELEMENT**  
**FIXER PARAMETRE ELEMENT**  
**LIRE PARAMETRE ELEMENT**  
**LISTE ENUMERATIONS**  
**SUPPRIMER DANS LISTE**

## Relatives aux ListBox

**INSERER COLONNE FORMULE LISTBOX**  
**FIXER TABLE SOURCE LISTBOX**  
**LIRE TABLE SOURCE LISTBOX**

## Relatives aux impressions

**OUVRIR TACHE IMPRESSION**  
**FERMER TACHE IMPRESSION**

## Relatives à l'environnement 4D :

**VERIFIER FICHER DONNEES**  
**VERIFIER FICHER DONNEES OUVERT**  
Compacter fichier donnees  
**OUVRIR CENTRE DE SECURITE**  
Lire langue courante base  
**LISTE COMPOSANTS**

## Relatives à l'environnement système :

**Selectionner couleur RVB** : affiche la fenêtre système de sélection de couleur  
**LIRE FORMATAGE SYSTEME** pour récupérer les valeurs des paramètres régionaux

## Relatives à l'interface utilisateur :

**Hauteur barre outils**

## Relatives aux méthodes

**EXECUTER METHODE** avec paramètres

## Relatives aux chaînes de caractères, XLIFF et regex

**CONVERTIR DEPUIS TEXTE**  
Convertir vers texte  
**Lire traduction chaîne** (uniquement pour le XLIFF)  
**Trouver regex**

## Relatives à la définition de structure

**Est un numero de table valide**  
**Est un numero de champ valide**  
**CREER INDEX**  
**SUPPRIMER INDEX**

## Relatives au Web

[Valider mot de passe digest Web](#) (pour la [Sur authentification Web](#) en mode digest)

## Relatives au XML

[DOM Chercher element XML par ID](#)  
[DOM EXPORTER VERS IMAGE](#)

## relatives aux images

[LISTE CODECS IMAGE](#)  
[TRANSFORMER IMAGE](#) (reprend et améliore les fonctionnalités des opérateurs sur les images)  
[COMBINER IMAGES](#) (d°)  
[CONVERTIR IMAGE](#)

Format images natifs

## Relatives aux Outils :

[Choisir](#) (criteres ; valeur1{ ;valeur N}) booléen ou numérique  
Exemple :  
VTitre :=[Choisir](#)([Contact]Masculin ; "Mr" ; "Madame")

[LIRE PARAMETRE MACRO](#)  
[FIXER PARAMETRE MACRO](#)

## Les commandes dont la syntaxe a été modifiée :

### Relatives aux menus :

Toutes les commandes relatives aux menus acceptent désormais une chaîne de type *RefMenu* dans le paramètre *Menu* et permettent de travailler avec les sous-menus hiérarchiques.

[Menu choisi permet de travailler avec les menus hiérarchiques](#)

### Relatives aux listes hiérarchiques :

Les commandes acceptent la syntaxe objet, syntaxe obligatoire en cas de multi-représentation d'une liste.

L'utilisation des commandes 4D standards ([CHANGER JEU DE CARACTERES](#), [CHANGER STYLE](#), [CHANGER TAILLE](#), [CHOIX VISIBLE BARRES DEFILEMENT](#), [DEFILER LIGNES](#), [CHOIX COULEUR](#), [FIXER COULEUR RVB](#), [CHOIX FILTRE SAISIE](#), [CHOIX SAISSABLE](#)) sur les listes hiérarchiques, en utilisant le nom d'objet ou le nom de variable, a été étendue.

### Relatives aux ListBox

[INSERER COLONNE LISTBOX](#)  
[INSERER LIGNE LISTBOX](#)  
[SUPPRIMER LIGNE LISTBOX](#)  
[LIRE TABLEAU LISTBOX](#)

[Objet focus](#) dans le contexte d'une listbox :

colonne associée à un champ : pointe sur le champ

Colonne associée à une variable : pointe vers la variable

Colonne associée à une expression : pointe vers la variable de la listbox

[Numero de ligne affichee](#) : fonctionne désormais avec une ListBox

### Relatives aux impressions

[PARAMETRES IMPRESSION](#)(typeDial)  
TypeDial : 0 toutes, 1 : format d'impression, 2 : impression

**Relatives aux composants :**

**Fichier structure** (\*) : retourne le fichier de structure de la base hôte (pour un composant)

Dossier 4D(dossier ;{\*}) avec \* retourne le dossier de la base hôte

Mode compilé (\*) information sur la base hôte

**Relatives à l'environnement de 4D :****FIXER PARAMETRE BASE****Lire parametre base**

<b>Nouveaux sélecteurs :</b>	<b>Sélecteurs modifiés :</b>	<b>Sélecteurs supprimés :</b>
41 : Mode unicode 42 : taille mémoire temporaire 43 : SQL autocommit	17 : jeu de caractère en mode unicode, utilise les jeux de caractères définis par l'IANA  29 : enregistrement requêtes web 30 : client enreg. Requêtes web avec nouveaux formats de fichier d'historique (CLF, DLF, ELF, WLF)	les sélecteur 1 ratio de tri séq 2 Optimisation accès séq, 3 ratio valeurs distinctes séq, 4 compression index, 5 ratio chercher dans sélec séq 26 mode écriture cache

**OUVRIR PREFERENCES 4D**

<b>Clés 4D 2004 :</b>	<b>Modifications ou Nouvelles clés</b>
/Application/Compatibility/Structure Compatibility	/Application/Compatibility/Design Compatibility
/Design Mode/Method Editor/Styles for Syntax Elements	/Design Mode/Method Editor/Syntax Styles
/Database/Script Manager	/Database/International
/Client-Server/Publishing/Allow-Deny Table Configuration	/Client-Server/Publishing/Allow-Deny Configuration Table
	/Application/Access/General Settings
	/Design Mode/Structure/Automatic Form Creation
	/Moving
	/Moving/Default Actions during the Copy if Dependent Objects
	/Moving/Moving Dialog
	/Database/International/Right-to-left Languages
	/Database/International/Numeric Display Format
	/Web/Options/Options
	/Web/Log Format
	/Web/Log Format/Web Log Type
	/Web/Log Format/Web Log Token Selection
	/Web/Log Scheduler
	/Web/Log Scheduler/Backup Frequency for Web Log File
	/SQL
	/SQL/Configuration
	/SQL/Configuration/SQL Server Access

**Relatives à l'environnement système :****PROPRIETES PLATE FORME**

<b>Constantes 4D 2004</b>	<b>Nouvelles constantes</b>
---------------------------	-----------------------------



Power Macintosh	Mac OS
Windows	Windows
Autres G3 et supérieurs	PowerPC
Pentium	Compatible Intel
INTEL 386	
INTEL 486	
Macintosh 68K	
PowerPC 601	
PowerPC 603	
PowerPC 604	
PowerPC G3	

## Relatives à la saisie et l'interface utilisateur :

**DIALOGUE** : nouveau paramètre \* le formulaire est alors chargé dans la dernière fenêtre ouverte du process courant (permet d'avoir une palette flottante sans lancer un nouveau process).

**Créer fenêtre** : nouveau paramètre Aspect metal sous MacOS

**CHOIX FORMATAGE**(\$MonThermometre ; ";;;128")

- Permet d'activer le mode **Barber shop** par programmation pour les thermomètres (mode 128)

**CHOIX FORMATAGE**(\$MonBouton ; ;#MesImages/MonImage.jpg)

- Permet de référencer, associée à un bouton 3D, une image .jpg se trouvant dans le sous-dossier "MesImages" du dossier "ressources" de la base

**ALLER A CHAMP**(\* ;"") plus aucun objet n'a le focus

## Relatives aux chaînes de caractères :

**Num** accepte désormais un nouveau paramètre, permettant de désigner un séparateur décimal. Par défaut, la commande utilise le séparateur décimal défini dans le système d'exploitation. Lorsque ce paramètre est passé, la commande ne tient pas compte du séparateur décimal système. Vous pouvez passer un ou plusieurs caractères.

**Chaîne** accepte désormais les alpha et les booléens (« vrai » « faux ») en paramètres

- **Minusc, Majusc**

Nouveau paramètre \* pour spécifier que l'on garde les accents

**Minusc**(chaîne{ ;\*})

**Majusc**(chaîne{ ;\*})

En 2004.x

**Majusc** ("é") -> E : accent perdu

**Minusc** ("É") -> è : accent conservé

En 4D v11

**Majusc** ("é") -> E : accent perdu (idem 2004)

**Minusc** ("É") -> e : accent perdu

**Majusc** ("é";\*) -> È : accent conservé

**Minusc** ("É";\*) -> è : accent conservé (idem 2004)

Exemple :

**Minusc** (« ÉÚÂÆ éABC») -> euæ eabc les accents ont disparu et tout est passé en minuscules

**Minusc** («ÉÚÂÆ éABC» ; \*) -> éúâæ éabc les accents sont préservés et tout est passé en minuscules

**Majusc** («èà ») -> EA : les accents ont disparus

**Majusc** («èà » ; \*) -> ÉÀ : les accents sont préservés

- **Position**

De nouveaux paramètres comme point de départ et \* ont été ajoutés. \* permet de spécifier que l'on recherche en diacritique.

Position a été réécrit et utilise la librairie **ICU** (International Components for Unicode) qui entraîne une incompatibilité car désormais Position ignore tous les caractères ignorés par Unicode comme le NULL ASCII ou Caractere(1).

**Attention** : pensez à utiliser le nouveau paramètre \* pour garder le même fonctionnement qu'en 4D 2004 :

\$p:=**Position**(Caractere(0);vTexte) \$p -> 1

Pour récupérer 0, il faut utiliser le caractère \*

\$p:=**Position**(Caractere(0);vTexte;\*) \$p -> 0

**A noter** : en Unicode, les **codes de caractères suivants** sont réservés et ne doivent jamais être inclus dans un texte :

0

65534 (FFFE)

65535 (FFFF)

## Relatives à la définition de structure :

**Champ** (si pointeur sur sous-champ, retourne le numéro ou le nom du sous-champ)

**Nom de champ** (d°)

**FIXER INDEX** (tient compte des nouveaux types d'index)

## Relatives au Web

**FIXER RACINE HTML** (désormais la commande ne tient pas compte des Préférences) – voir Modifications de comportements.

## Relatives au XML

**DOM Chercher element XML**

**DOM Analyser source XML**

**DOM Analyser variable XML**

**SAX AJOUTER VALEUR ELEMENT XML**

**APPLIQUER TRANSFORMATION XSLT**

## Relatives aux Documents système

**Sélectionner dossier** : nouveau paramètre *CheminDefaut*

**Ouvrir document**(document{; type{; mode}})

**Créer document**

**Ajouter a document**

Le paramètre *type* a été modifié pour ces trois commandes. Il est désormais possible de passer dans ce paramètre une liste de types de documents, séparés par des points-virgules (;).

## Relatives aux BIObs

**TEXTE VERS BLOB**

**BLOB VERS TEXTE**

## Relatives aux ressources

**Lire chaine dans liste**

**LISTE DE CHAINE VERS TABLEAU**

Prennent en charge le XLIFF

**Relatives aux communications :**

**ENVOYER PAQUET** (autorise les blobs)  
**RECEVOIR PAQUET** (autorise les blobs)  
**UTILISER FILTRE** (en Unicode, nom IANA du jeu de caractères)

**Relatives aux Web services**

**APPELER WEB SERVICE** (Keep-alive possible avec nouveau paramètre \*)

**Relatives au process**

Nouvelles constantes origine :

Process du serveur Web	Entier long	-13
Process exécuté sur client	Entier long	-14
Process 4D Server	Entier long	-15
Process sur fermeture	Entier long	-16
Process macro éditeur de method	Entier long	-17
Process 4D Server interne	Entier long	-18
Process de sauvegarde	Entier long	-19
Process du fichier d'historique	Entier long	-20
Process de restitution	Entier long	-21
Process CSM	Entier long	-22
Process minuteur	Entier long	-23
Process exécution méthode SQL	Entier long	-24
Process Server Controller	Entier long	-25
Process d'activité	Entier long	-26

**Les commandes dont le nom a été modifié :****Thèmes divers**

**EXECUTER**  
**TROUVER CLEF INDEX**

devient

**EXECUTER FORMULE**  
**TROUVER DANS CHAMP**

fonctionne sur champ non-indexé et @ peut être utilisé dans la chaîne (mode séquentiel forcé dans ce cas).

**Code ascii**  
**LIRE LISTE PLUGINS**

**Code de caractere**  
**LISTE PLUGINS**

**Relatives au conteneur de données :**

**Tester presse papiers**  
**LIRE PRESSE PAPIERS**  
**EFFACER PRESSE PAPIERS**  
**AJOUTER A PRESSE PAPIERS**  
**ECRIRE IMAGE DANS PRESSE PAPIERS**  
**LIRE IMAGE DANS PRESSE PAPIERS**  
**ECRIRE TEXTE DANS PRESSE PAPIERS**  
**Lire texte dans presse papiers**

devient

**Tester conteneur**  
**LIRE DONNEES CONTENEUR**  
**EFFACER CONTENEUR**  
**AJOUTER DONNEES AU CONTENEUR**  
**FIXER IMAGE DANS CONTENEUR**  
**LIRE IMAGE DANS CONTENEUR**  
**FIXER TEXTE DANS CONTENEUR**  
**Lire texte dans conteneur**

**Relatives au Menus :**

**CHANGER BARRE**  
**Titre menu**  
**Texte ligne menu**  
**CHANGER TEXTE LIGNE MENU**

devient

**FIXER BARRE MENUS**  
**lire titre menu**  
**Lire texte ligne menu**  
**FIXER TEXTE LIGNE MENU**

Style ligne menu  
**CHANGER STYLE LIGNE MENU**  
 Marque ligne menu  
**MARQUER LIGNE MENU**  
**CHANGER RACCOURCI CLAVIER**

Lire style ligne menu  
**FIXER STYLE LIGNE MENU**  
 Lire marque ligne menu  
**FIXER MARQUE LIGNE MENU**  
**FIXER RACCOURCI LIGNE MENU**

### Relatives aux listes hiérarchiques

**SUPPRIMER ELEMENT**  
**INSERER ELEMENT**

devient

**SUPPRIMER DANS LISTE**  
**INSERER DANS LISTE**

### Relatives à la définition de structure

**Nombre de tables**  
**Nombre de champs**

devient

**Lire numero dernière table**  
**Lire numero dernier champ**

### Relatives à l'Import/export

**LECTURE ASCII**  
**ECRITURE ASCII**

devient

**IMPORTER TEXTE**  
**EXPORTER TEXTE**

### Relatives aux tableaux

**INSERER LIGNES**  
**SUPPRIMER LIGNES**

devient

**INSERER DANS TABLEAU**  
**SUPPRIMER DANS TABLEAU**

## Les commandes dont le comportement a été modifié :

- **TOUT SELECTIONNER** n'accepte plus un champ dépointé comme paramètre.
- **Type version** : en 4D 2004 renvoyée 0 pour la version standard et 1 pour la version démo. En 4D v11 SQL, cette fonction renvoie toujours 0, le mode d'évaluation de 4D ayant changé. Désormais en mode d'évaluation on a accès à tout (plug-in...) mais pour 1 mois seulement.
- **FIXER RACINE HTML** (*dossierRacine*)  
 Cette commande fonctionne désormais sans restriction en mode sans contexte. Elle modifie le dossier racine pour tous les process Web pendant la session courante. Le chemin du dossier racine HTML défini dans les Préférences n'est pas modifié. Le paramètre *dossierRacine* accepte donc désormais une chaîne du type :

Sous Mac OS Disque:Applications:monserv:dossier  
 Sous Windows C:\Applications\monserv\dossier

La syntaxe précédente (type "url" relatif) reste utilisable.

- **LIRE VARIABLES FORMULAIRE WEB** nécessite désormais la déclaration des tableaux utilisés
- **Creer document**  
 En 4D v11SQL, la variable **Document** renvoie le chemin d'accès complet, alors qu'en 4D 2004 elle renvoyée le nom indiqué lors du Créer document.

Exemple :

```
$Ref:= Creer document ("MonBeauDocument")
Document renvoie
v2004 -> MonBeauDocument
v11 SQL -> C:\MonDossier\Mabase\MonBeauDocument
```

- Gestion des ruptures en impression  
**CUMULER SUR** et **NIVEAUX DE RUPTURE** : désormais obligatoire en mode interprété (l'étaient déjà en mode compilé)  
**CUMULER SUR** fonctionne maintenant sur les variables interprocess  
 Pour les champs et les variables texte, seuls les 80 premiers caractères étaient pris en compte, avec 4D v11, c'est l'intégralité du texte qui est pris en compte.
- **EFFACER VARIABLE** : cette commande fonctionne désormais à l'identique en mode interprété et compilé. Réinitialise les variables à leur valeur par défaut dans les deux modes.  
 En 4D 2004 : en interprété, supprimait les variables de la mémoire, en compilé : réinitialisait)  
 N'accepte plus la syntaxe EFFACER VARIABLE(« toto »)
- **Champ** et **Nom du champ** : si vous passez un pointeur à un champ, ces fonctions retournent le n° et le nom du champ de la sous-table au lieu du champ source (c'était déjà le cas en 4D 2004 !!! mais non documenté).
- **Numero dans selection** : dans le cadre d'un **IMPRIMER ENREGISTREMENT**, renvoyait 0. en 4D v11, renvoie la vraie position.
- **STOCKER ENSEMBLE** : utilise un nouveau format : les ensembles stockés en 4D v11 ne sont pas lisibles par les anciennes versions, mais **CHARGER ENSEMBLE** relit l'ancien format.
- **ENVOYER ENREGISTREMENT** : utilise un nouveau format. Les enregistrements envoyés avec 4D v11 ne sont pas utilisables par les anciennes versions, mais **RECEVOIR ENREGISTREMENT** relit l'ancien format mais pas s'il contient des sous-tables. (rappel : les sous-tables ne sont plus supportées).
- **QUITTER 4D** fonctionne maintenant en mode asynchrone.
- **ENDORMIR PROCESS** marche maintenant sur le process de « UserMode »
- **SELECTION VERS TABLEAU** ne change plus des fiches courantes, les sélections de la table principale et des tables liées. Les liens automatiques sont activés. Les fiches courantes des tables liées ne sont plus déchargées.
- **TABLEAU VERS SELECTION** ignore les champs qui n'appartiennent pas à la première table. Renvoie une erreur si les tableaux ne sont pas de la même taille.
- **Self** : le pointeur **Self** dans une méthode-objet appelant une méthode-projet : en 2004 compilé renvoyait le pointeur sur l'objet, mais en interprété, renvoyait : Nil. En 4D v11 SQL, renvoie non nul en interprété et compilé.
- **JOINTURE** : en 2004 prenait le premier chemin possible. En 4D v11 SQL rend le chemin le plus optimisé : cela peut entraîner des incompatibilités : les résoudre en précisant dans la commande **JOINTURE** la liste des champs par lesquels il faut passer (Nouveautés 4D v11 SQL)
- Débordement avec indice [[i]] : message d'erreur (absent en en 4D 2004).
- **CHERCHER**  

[Table1]	[Table2]	[Table3]
Champ1->	Champ1	
	Champ2 ->	Champ1

**CHERCHER**([Table3];[Table1]Champ1=1) `quelque soit le type de champ  
 En 4D v2004 -> Pas d'enregistrement trouvés dans [Table3]  
 En 4D v11 SQL -> les enregistrements sont trouvés dans [Table3]
- Déchargement-rechargement de la fiche courante :  
**ALLER A ENREGISTREMENT** : sur la fiche courante ne la recharge que si le mode **LECTURE SEULEMENT/ LECTURE ECRITURE** a changé. Idem pour de chargement de la première fiche après un **CHERCHER**.

**JOINTURE** : 4D 2004 déchargeait l'enregistrement courant. 4D v11 SQL ne le fait plus.

- **Position déposer** : fonctionne désormais pour les variables, champ texte et combo box
- Pointeurs sur les variables process.  
Désormais le fonctionnement est identifié en interprété et en compilé
- Modifications concernant les Tableaux à 2 dimensions :
  - Si vous passez un tableau à deux dimensions en paramètre d'une commande qui attend une variable, 4D génère une erreur. Dans les versions précédentes, aucune erreur n'était générée mais la seconde dimension n'était pas prise en compte.
  - Lorsque la fonction **Type** est appliquée à une "ligne" d'un tableau 2D, elle retourne désormais le véritable type du tableau 2D et non plus le type "Est un tableau 2D".
  - Tableaux 2 dimensions alpha : il n'est plus possible de changer la longueur d'un alpha en mode interprété.

## **Commandes pour lesquelles le premier paramètre, la table, est maintenant obligatoire :**

**APPLIQUER A SELECTION**  
**CHERCHER PAR FORMULE**  
**CHERCHER PAR FORMULE DANS SELECTION**  
**EDITER FORMULE**

## **Les commandes qui changent de thèmes**

*Thème « Interface utilisateur »*  
**CACHER BARRE DE MENUS**  
**AFFICHER BARRE DE MENUS**  
**APPELER SUR A PROPOS**  
**AFFICHER BARRE OUTILS**  
**CACHER BARRE OUTILS**

*Thème « Environnement 4D »*  
**FIXER PARAMETRE BASE**  
**Lire parametre base**  
**GENERER APPLICATION**  
**LISTE PLUGINS**

*Thème « Outils »*  
**LANCER PROCESS EXTERNE**  
**LIRE VARIABLE ENVIRONNEMENT**

*Thème « Utilisateurs et groupes »*  
**Licence disponible**

*Thème « Formules »*  
**EXECUTER FORMULE**

## **Les commandes optimisées**

De nombreuses commandes ont été optimisées dans 4D v11 SQL. Plus particulièrement :

**CHERCHER PAR FORMULE**  
**CHERCHER PAR FORMULE DANS SELECTION**  
**CHERCHER PAR TABLEAU**

## Les commandes obsolètes

A remplacer par :

<b>CHERCHER SUR CLEF</b>	<b>CHERCHER</b>
<b>TRIER SUR INDEX</b>	<b>TRIER</b>
<b>STOCKER ANCIEN</b>	<b>STOCKER SUR LIEN</b>
<b>Lire ID ressource composant</b>	-
<b>LISTE SEGMENTS DE DONNEES</b>	-
<b>AJOUTER SEGMENT DE DONNEES</b>	-
<b>Type version</b>	-

en 4D 2004, cette fonction renvoyée 0 pour la version standard et 1 pour la version démo.

En 4D v11 SQL, cette fonction renvoie toujours 0, le mode d'évaluation de 4D ayant changé.

Désormais en mode d'évaluation, on a accès à tout (plug-in...) mais pour 1 mois seulement.

## Les commandes dont l'utilisation est devenue inutile

**REDESSINER LISTE** (liste hiérarchique)

## Les syntaxes obsolètes

- **Si (Numero enregistrement([Table])=-3)**

Cette syntaxe est déconseillée depuis 4D v 6.5. Dans le trigger Sur sauvegarde nouvel enregistrement, cette fonction ne retourne plus -3 mais le numéro d'enregistrement

A remplacer par :

**Si (Nouvel enregistrement([Table]))**

- Les syntaxes **Si(Avant)...** (**Pendant**)... (**Après**) dans les méthodes formulaires doivent être remplacées par un test sur l'évènement formulaire : **Si(Evenement formulaire= Sur chargement)** etc...
- **SUPPRIMER LISTE** (Liste{ ;\*}) utilisée pour supprimer les sous-listes est à remplacer par la commande **SUPPRIMER DANS LISTE** (qui est l'ancienne commande **SUPPRIMER ELEMENT** renommée).
- **\$Structure :=fichier structure**  
**\$Ressources :=Ouvrir fichier ressources** (\$Structure)

## Les commandes sur les sous-enregistrements

Après conversion d'une sous-table en table standard, les commandes sur les sous-enregistrements continuent de fonctionner tant que vous ne supprimez pas le lien spécial entre les deux tables.

Si vous le supprimez, il faudra utiliser les commandes concernant les tables.

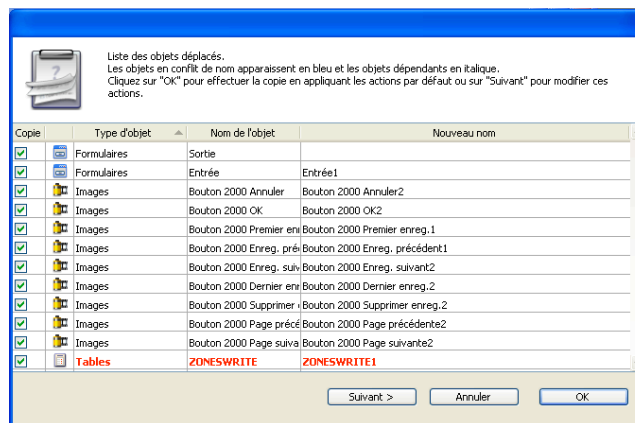
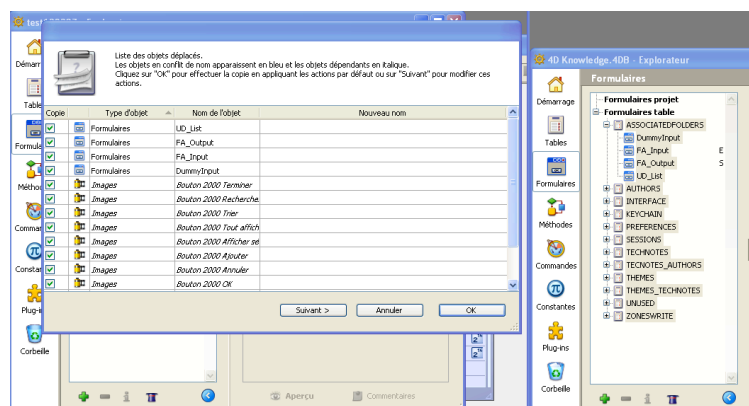
Attention : Les commandes comme **DUPLIQUER ENREGISTREMENT**, **ENVOYER ENREGISTREMENT** Et **RECEVOIR ENREGISTREMENT** ne tiennent plus compte des sous-tables.

## Les intégrations (partielles)

## 4D Insider remplacé par le « Glisser-déposer système » et par le « Chercher-remplacer » :

Peuvent être déplacés d'une base à l'autre : (intégration partielle 4D Insider)

- Enumération
- Feuilles de style
- Format et filtre
- Images de la bibliothèque d'image
- Barre de menus et menus
- Infobulles
- Formulaire projet avec méthode formulaire
- Formulaire table avec méthode formulaire
- Méthode projet
- Dossier/sous dossier
- Table avec champs et trigger
- Tous les objets d'un formulaire (avec méthode objet – 4D 2004)



Les objets en conflit de nom apparaissent en rouge.  
Les objets dépendants en italique

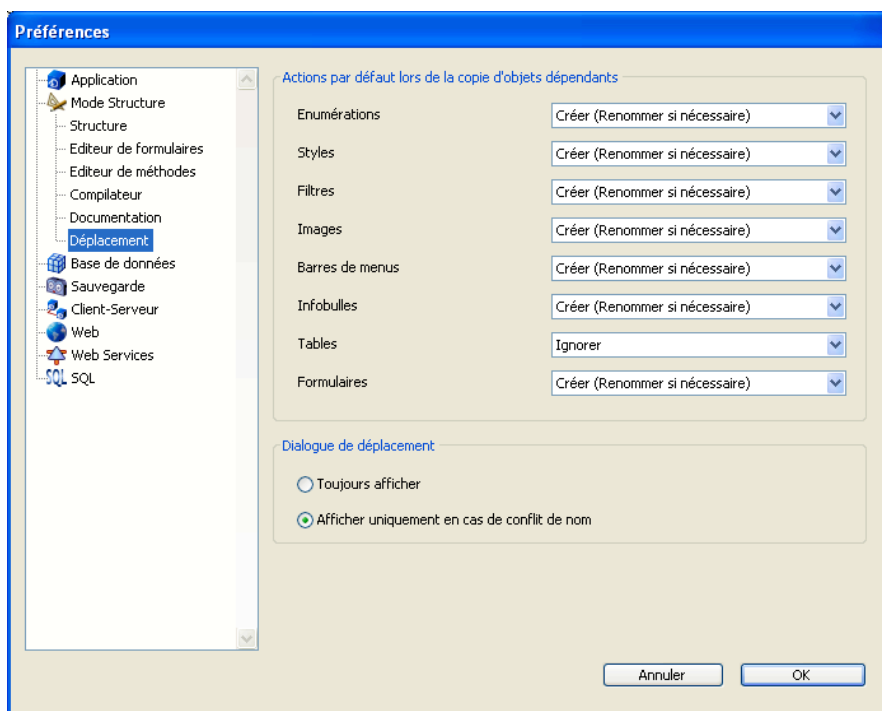
**Dans un menu, la référence à la méthode projet est conservée. L'héritage de formulaire est aussi conservé.**

Les droit d'accès ne sont pas gardés.

ALT CLIC (win) OPTION CLIC (mac) pour tout sélectionner et désélectionner.  
Utiliser un autre objet : n'existe que pour les objets dépendants

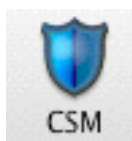
Vous pouvez définir les options de déplacement dans les Préférences :





## 4D Tools remplacé par le CSM

Le Centre de Sécurité et de maintenance remplace désormais 4D Tools. Il est intégré dans 4D et vous y accéder par la palette d'outils :



## 4D Pack

Avec 4D V11 SQL, et suite à des problèmes de migration liés à la disparition de certaines commandes de 4D Pack, le chargement de 4D Pack 2004 est autorisé pour une migration plus facile. Nous dégageons toutefois toute responsabilité quant au fonctionnement de 4D Pack 2004 en v11.

## **Commandes 4D Pack obsolètes**

Après conversion de bases de données existantes, ces commandes doivent être remplacées :

Commandes 4D Pack :	A remplacer par :
<b>AP PICT DRAGGER</b> <b>AP PICT UPDATER</b>	Glisser-déposer 4D zone externe 4D Pack %AP Pict displayer à remplacer par un champ image
<b>AP Read Picture File</b>	<b>LIRE FICHIER IMAGE</b>
<b>AP Save BMP 8 bits,</b> <b>AP Save GIF</b>	<b>CONVERTIR IMAGE</b> et <b>ECRIRE FICHIER IMAGE</b>
<b>AP Read Picture BLOB,</b>	inutile, les images étant stockés en natif Ou utiliser : <b>BLOB VERS IMAGE</b>
<b>AP PICT DRAGGER</b> <b>AP Rect dragger</b>	variables système <b>MouseDown</b> , <b>MouseX</b> et <b>MouseY</b>
<b>AP SET PICT MODE</b>	utiliser : <b>CHOIX FORMATAGE</b>
<b>AP Select document</b>	utiliser : <b>Selectionner document</b>
<b>AP Set palette</b>	utiliser : <b>Selectionner couleur RVB</b>
<b>AP ShellExecute</b>	utiliser : <b>LANCER PROCESS EXTERNE</b>
<b>AP Sublaunch</b>	utiliser : <b>LANCER PROCESS EXTERNE</b>
<b>Déjà obsolète en 4D 2004 :</b>	
<b>AP PrintDefault</b> <b>AP Text to PrintRec</b> <b>AP PrintRec to text</b> <b>AP PrValidate</b>	Utiliser : <b>AP Blob to print setting</b> et <b>AP Print settings to Blob</b>
<b>AP Toolbar installed</b>	utiliser : <b>Hauteur barre outils</b> (retourne 0 si non installée)
<b>AP SET WEB FILTERS</b>	utiliser : <b>FIXER PARAMETRES BASE</b>
<b>AP Add table and fields</b>	utiliser les nouvelles capacités SQL de 4D v11 SQL <b>exemple :</b> <b>CREATE TABLE Client</b> <b>(Nom char(50),</b> <b>Prenom char(50),</b> <b>Adresse char(50),</b> <b>Ville char(50),</b> <b>Pays char(25),</b> <b>Date_naissance date)</b>
<b>AP Create relation</b>	<b>pas de contournement pour l'instant (implémentation prévue pour une future version)</b>

**Commandes toujours présentes mais qu'il est conseillé de ne plus utiliser :**

**AP Save BMP 8 bits**

**AP Get picture type**

**AP GET PARAM**  
**AP SET PARAM** beaucoup de paramètres sont obsolètes et peuvent être remplacés par **FIXER PARAMETRES BASE**

**AP AVAILABLE MEMORY** : attention les valeurs sont maintenant en Ko.

**AP FCLOSE**  
**AP fopen**  
**AP FPRINT**  
**AP fread** peuvent être remplacés par **REGLER SERIE** et **ENVOYER PAQUET** et **RECEVOIR PAQUET**

**AP CLOSE HELP**  
**AP HELP INDEX**  
**AP HELP ON HELP**  
**AP HELP ON KEY** Ne fonctionne que sous Windows XP, pas compatible VISTA

**Commande améliorée****AP Create method**

Le paramètre *TabPropriétés* peut contenir 2 valeurs de plus :  
- passez 1 dans *TabPropriétés* {5} si la méthode doit être partagée entre un composant et une base hôte.  
- passez 1 dans *TabPropriétés* {6} si la méthode doit être utilisée en contexte SQL.

**Commandes identiques**

**AP Get templates**  
**AP Rect dragger**  
**AP SET CLOCK**  
**AP Timestamp to GMT**  
**AP FULL SCREEN**  
**AP NORMAL SCREEN**  
**AP BLOB to print settings**  
**AP Print settings to BLOB**  
**AP Get tips state**  
**AP SET TIPS STATE**  
**AP Get field infos**  
**AP Get table info**

**Nouvelles commandes**

- **AP Get File MD5 Digest** (filePath;Digest;Resfork) -> Erreur

Paramètre	Type		Description
filePath	Texte	->	chemin accès complet du fichier
Value2	Entier long	<-	MD5 digest
resFork	Entier long	->	0 : Datafork ou 1 : Resource fork

Description  
La commande **AP Get File MD5 Digest** retourne la clef MD5 d'un document

**C\_TEXTE**(\$Document)  
**C\_ENTIER LONG**(<>Digest ;\$resfork)

```
$Resfork:=0  
$Document:=Selectionner document("" ; "" ; "" ;0)  
$Error:= AP Get File MD5 Digest ($Document;<>Digest;$Resfork)
```

- **AP Does method exist**

# Les Raccourcis clavier

**ALT à l'ouverture** d'une base pour créer ou choisir un fichier de données

## Développement

Imprimer : CTRL/CMD P  
Format Impression : CTRL/CMD SHIFT P  
Ouvrir un formulaire : CTRL/CMD L  
Ouvrir l'explorateur : CTRL/CMD SHIFT E  
Ouvrir la boîte à outils : CTRL/CMD SHIFT T  
Compilateur : CTRL/CMD \*  
Chercher dans le développement : CTRL/CMD SHIFT F  
Préférences : CTRL/CMD /

Tester application : CTRL/CMD I  
Exécuter méthoder : CTRL/CMD R  
Lancer interprété CTRL/CMD ALT I  
Lancer compilé : CTRL/CMD SHIFT I  
Minimisé fenêtre : CTRL/CMD M  
Minimisé toutes les fenêtre : CTRL/CMD ALT SHIFT M

### **Structure :**

Tout sélectionner (table ou c hamp) : CTRL/CMD A  
Sélection continue ou discontinue (table ou champ) : CLIC SHIFT CLIC ou CLIC CTRL/CMD CLIC  
TAB et MAJ TAB pour sélectionner Table suivante et Table précédente  
Retour chariot pour créer un champ

### **Méthode**

Ouvrir une méthode : CTRL/CMD K  
Dans l'éditeur de méthode, ouvrir une méthode : ALT DOUBLE CLIC  
numéro de ligne : ALT SHIFT N  
Prochaine erreur : CTRL/CMD =  
Erreur précédente : CTRL/CMD ALT =  
Sélectionnez bloc : CTRL/CMD B  
Commenté/ décommenté : CTRL/CMD /  
Intervertir l'expression : CTRL/CMD ALT T

### **Formulaire**

Déplacer au premier plan : CTRL/CMD F  
Déplacer en arrière-plan : CTRL/CMD B  
Grouper : CTRL/CMD G  
Dégrouper : CTRL/CMD SHIFT G  
Infobulle sur l'objet : CTRL/CMD SHIFT sur l'objet  
Accéder à un objet sur page 0 à partir d'une autre page : CTRL/CMD CLIC sur l'objet

## Enregistrement

Liste des tables : CTRL/CMD SHIFT U  
Afficher la table courante : CTRL/CMD U  
Nouvel enregistrement en liste : CTRL/CMD SHIFT N  
Nouvel enregistrement en page : CTRL/CMD ALT N  
Tout montrer : CTRL/CMD +  
Sous sélection : CTRL/CMD –  
Trier : CTRL/CMD SHIFT Y  
Appliquer une formule : CTRL/CMD SHIFT R  
Chercher : CTRL/CMD Y  
Ecrire cache données : CTRL/CMD ALT SHIFT S



## Annexe 1 : Les codes langues:

Les codes langues sont les suivants (tels que mentionnés dans le fichier **keyboardmapping.xml**).

DC\_AFRIKAANS = 1078  
DC\_ALBANIAN = 1052  
DC\_ARABIC\_SAUDI\_ARABIA = 1025  
DC\_ARABIC\_IRAQ = 2049  
DC\_ARABIC\_EGYPT = 3073  
DC\_ARABIC\_LIBYA = 4097  
DC\_ARABIC\_ALGERIA = 5121  
DC\_ARABIC\_MOROCCO = 6145  
DC\_ARABIC\_TUNISIA = 7169  
DC\_ARABIC\_OMAN = 8193  
DC\_ARABIC\_YEMEN = 9217  
DC\_ARABIC\_SYRIA = 10241  
DC\_ARABIC\_JORDAN = 11265  
DC\_ARABIC\_LEBANON = 12289  
DC\_ARABIC\_KUWAIT = 13313  
DC\_ARABIC\_UAE = 14337  
DC\_ARABIC\_BAHRAIN = 15361  
DC\_ARABIC\_QATAR = 16385  
DC\_BASQUE = 1069  
DC\_BELARUSIAN = 1059  
DC\_BULGARIAN = 1026  
DC\_CATALAN = 1027  
DC\_CHINESE\_TRADITIONAL = 1028  
DC\_CHINESE\_SIMPLIFIED = 2052  
DC\_CHINESE\_HONGKONG = 3076  
DC\_CHINESE\_SINGAPORE = 4100  
DC\_CROATIAN = 1050  
DC\_CZECH = 1029  
DC\_DANISH = 1030  
DC\_DUTCH = 1043  
DC\_DUTCH\_BELGIAN = 2067  
DC\_ENGLISH\_US = 1033  
DC\_ENGLISH\_UK = 2057  
DC\_ENGLISH\_AUSTRALIA = 3081  
DC\_ENGLISH\_CANADA = 4105  
DC\_ENGLISH\_NEWZEALAND = 5129  
DC\_ENGLISH\_EIRE = 6153  
DC\_ENGLISH\_SOUTH\_AFRICA = 7177  
DC\_ENGLISH\_JAMAICA = 8201  
DC\_ENGLISH\_CARIBBEAN = 9225  
DC\_ENGLISH\_BELIZE = 10249  
DC\_ENGLISH\_TRINIDAD = 11273  
DC\_ESTONIAN = 1061  
DC\_FAEROESE = 1080  
DC\_FARSI = 1065  
DC\_FINNISH = 1035  
DC\_FRENCH = 1036  
DC\_FRENCH\_BELGIAN = 2060  
DC\_FRENCH\_CANADIAN = 3084  
DC\_FRENCH\_SWISS = 4108  
DC\_FRENCH\_LUXEMBOURG = 5132  
DC\_GERMAN = 1031  
DC\_GERMAN\_SWISS = 2055  
DC\_GERMAN\_AUSTRIAN = 3079

DC\_GERMAN\_LUXEMBOURG = 4103  
DC\_GERMAN\_LIECHTENSTEIN = 5127  
DC\_GREEK = 1032  
DC\_HEBREW = 1037  
DC\_HUNGARIAN = 1038  
DC\_ICELANDIC = 1039  
DC\_INDONESIAN = 1057  
DC\_ITALIAN = 1040  
DC\_ITALIAN\_SWISS = 2064  
DC\_JAPANESE = 1041  
DC\_KOREAN\_WANSUNG = 1042  
DC\_KOREAN\_JOHAB = 2066  
DC\_LATVIAN = 1062  
DC\_LITHUANIAN = 1063  
DC\_NORWEGIAN = 1044  
DC\_NORWEGIAN\_NYNORSK = 2068  
DC\_POLISH = 1045  
DC\_PORTUGUESE = 2070  
DC\_PORTUGUESE\_BRAZILIAN = 1046  
DC\_ROMANIAN = 1048  
DC\_RUSSIAN = 1049  
DC\_SERBIAN\_LATIN = 2074  
DC\_SERBIAN\_CYRILLIC = 3098  
DC\_SLOVAK = 1051  
DC\_SLOVENIAN = 1060  
DC\_SPANISH\_CASTILLAN = 1034  
DC\_SPANISH\_MEXICAN = 2058  
DC\_SPANISH\_MODERN = 3082  
DC\_SPANISH\_GUATEMALA = 4106  
DC\_SPANISH\_COSTA\_RICA = 5130  
DC\_SPANISH\_PANAMA = 6154  
DC\_SPANISH\_DOMINICAN\_REPUBLIC = 7178  
DC\_SPANISH\_VENEZUELA = 8202  
DC\_SPANISH\_COLOMBIA = 9226  
DC\_SPANISH\_PERU = 10250  
DC\_SPANISH\_ARGENTINA = 11274  
DC\_SPANISH\_ECUADOR = 12298  
DC\_SPANISH\_CHILE = 13322  
DC\_SPANISH\_URUGUAY = 14346  
DC\_SPANISH\_PARAGUAY = 15370  
DC\_SPANISH\_BOLIVIA = 16394  
DC\_SPANISH\_EL\_SALVADOR = 17418  
DC\_SPANISH\_HONDURAS = 18442  
DC\_SPANISH\_NICARAGUA = 19466  
DC\_SPANISH\_PUERTO\_RICO = 20490  
DC\_SWEDISH = 1053  
DC\_SWEDISH\_FINLAND = 2077  
DC\_THAI = 1054  
DC\_TURKISH = 1055  
DC\_UKRAINIAN = 1058  
DC\_VIETNAMESE = 1066