

# 4<sup>e</sup> Dimension 2004

---

*Mise à jour*  
*Windows<sup>®</sup>/Mac<sup>™</sup> OS*



---

## **4<sup>e</sup> Dimension 2004**

### **Mise à jour**

Copyright© 1985 - 2004 4D SA / 4D, Inc.  
Tous droits réservés.

---

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

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

4D, 4D Draw, 4D Write, 4D View, 4D Insider, 4<sup>e</sup>me Dimension®, 4D Server ainsi que les logos 4e Dimension et 4D sont des marques enregistrées de 4D SA.

Windows, Windows NT et Microsoft sont des marques enregistrées de Microsoft Corporation.

Apple, Macintosh, Power Macintosh, QuickTime sont des marques enregistrées ou des noms commerciaux de Apple Computer, Inc.

Mac2Win Software est un produit de Altura Software, Inc.

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

Correcteur orthographique, © Copyright SYNAPSE Développement, Toulouse, France, 1994-2004.

ACROBAT © Copyright 1987-2004, Secret Commercial Adobe Systems Inc. Tous droits réservés. ACROBAT est une marque enregistrée d'Adobe Systems Inc.

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

# Sommaire

<b>Chapitre 1</b>	<b>Introduction. . . . .</b>	<b>13</b>
	Présentation . . . . .	13
	Contenu de ce manuel . . . . .	14
	Compatibilité avec les versions précédentes . . . . .	15
	Configuration minimale. . . . .	16
<b>Chapitre 2</b>	<b>Architecture des applications 4D . . . . .</b>	<b>17</b>
	4 <sup>e</sup> Dimension au format Mach-O (MacOS) . . . . .	17
	Qu'est-ce que le format Mach-O ? . . . . .	17
	Mach-O et MacOS 9 . . . . .	17
	Compatibilité des plug-ins . . . . .	18
	Nouvelle architecture des plug-ins . . . . .	18
	Emplacements des les plug-ins . . . . .	18
	Architecture interne des plug-ins . . . . .	21
	Command Line Interface (CLI) sous MacOS X . . . . .	23
	Principes d'utilisation . . . . .	23
	Commandes et arguments . . . . .	24
	Enregistrer 4D Server comme service sous MacOS X . . . . .	26
	Fichier d'aide syntaxique . . . . .	27
<b>Chapitre 3</b>	<b>Mode Utilisation . . . . .</b>	<b>29</b>
	Editeur de formulaires utilisateurs. . . . .	29
	Description de l'éditeur . . . . .	29
	Visualisation et modification des objets . . . . .	31
	Accès aux bibliothèques d'objets . . . . .	34
	Editeur de formules . . . . .	35
	Interface . . . . .	35
	Liste des commandes . . . . .	37
	Importation et exportation . . . . .	37
	Appliquer la transformation XSL . . . . .	38
	Importation et exportation via une source de données ODBC . . . . .	38

Editeur d'états rapides . . . . .	45
Génération du code . . . . .	45
Ecart type. . . . .	48

## Chapitre 4

### **Mode Structure . . . . . .49**

Boîte de dialogue d'identification . . . . .	50
Modification du mot de passe . . . . .	50
Sauvegarde des paramètres d'accès à la base. . . . .	52
Préférences . . . . .	54
Tableau de comparaison . . . . .	55
Nouvelles options de compatibilité . . . . .	58
Editeur de formulaires . . . . .	60
Gestion de la mémoire cache . . . . .	61
Reconnexion automatique des clients. . . . .	64
Nouveautés provenant de 4D Customizer Plus . . . . .	65
Interface . . . . .	71
Barres d'outils . . . . .	71
Réorganisation des menus . . . . .	72
Inspecteur . . . . .	76
Explorateur d'exécution . . . . .	77
Boîte à outils . . . . .	77
Compilateur . . . . .	89
Explorateur . . . . .	90
Page Corbeille . . . . .	91
Affichage détaillé dans la page Composants. . . . .	95
Prévisualisation des méthodes et affichage des propriétés . . . . .	95
Utiliser des dossiers dans l'Explorateur . . . . .	96
Présentation . . . . .	97
Dossiers par défaut . . . . .	99
Créer des dossiers ou des sous-dossiers . . . . .	99
Supprimer un objet . . . . .	101
Organiser les dossiers et les sous-dossiers . . . . .	102
Informations sur le dossier. . . . .	105
Correspondance avec les groupes de 4D Insider. . . . .	105
Editeur de méthodes . . . . .	106
Macro d'appel de code 4D . . . . .	106
Nouveau dossier Macros . . . . .	109
Nouvelle barre d'outils . . . . .	110

---

Editeur de formulaires . . . . .	111
Palette et Liste des propriétés . . . . .	112
Barres d'objets et d'outils intégrées . . . . .	115
Paramètres d'impression des formulaires . . . . .	120
Zoom . . . . .	120
Gestion des vues. . . . .	121
Utilisation des badges . . . . .	125
Utiliser comme modèle . . . . .	128
Sélection d'objets groupés ou à l'arrière-plan . . . . .	129
Utiliser la grille magnétique. . . . .	129
Modifiable par l'utilisateur . . . . .	130
Ajustement dynamique . . . . .	132
Objets de formulaires . . . . .	133
List Box . . . . .	133
Gestion des séparateurs . . . . .	146
Interface de plate-forme. . . . .	148
Contrôles de saisie natifs . . . . .	150
Correction orthographique . . . . .	152
Cases à cocher à trois états . . . . .	154
Modification des boutons 3D . . . . .	155
Autres modifications . . . . .	163
Gestion des listes d'enregistrements . . . . .	165
Modes de sélection . . . . .	166
Saisie en liste. . . . .	168
Affichage des listes . . . . .	169
Options des sous-formulaires . . . . .	172
Actions standard. . . . .	174
Bibliothèques d'objets . . . . .	175
Créer une bibliothèque d'objets. . . . .	175
Ouvrir une bibliothèque d'objets . . . . .	176
Construire une bibliothèque d'objets. . . . .	176
Affichage des bibliothèques d'objets . . . . .	179
Fonctionnement en client/serveur . . . . .	180
Générateur d'applications . . . . .	181
Nouveaux noms pour les applications de déploiement . . . . .	181
Boîte de dialogue et projet de génération d'application . . . . .	182
Construction d'une application client/serveur . . . . .	184
Gestion des plug-ins. . . . .	189
Gestion des numéros de licence. . . . .	190
Mise à jour automatique des applications clientes . . . . .	192
Web Services . . . . .	194
Publication en mode DOC (serveur) . . . . .	194
Utilisation des Web Services (client) . . . . .	195

<b>Chapitre 5</b>	<b>Module de sauvegarde intégré . . . . .</b>	<b>197</b>
	Présentation . . . . .	197
	Gestion des sauvegardes . . . . .	198
	Effectuer une sauvegarde. . . . .	198
	Configuration des fichiers de sauvegarde . . . . .	203
	Paramètres de sauvegarde . . . . .	208
	Paramétrer des sauvegardes périodiques . . . . .	214
	Sauvegarde automatique . . . . .	216
	Gestion du fichier d'historique . . . . .	217
	Présentation . . . . .	218
	Créer le fichier d'historique . . . . .	220
	Fermer l'historique . . . . .	223
	Analyser l'historique . . . . .	223
	Récupération des données . . . . .	227
	Incidents et diagnostic . . . . .	227
	Réparations automatiques . . . . .	228
	Restitution manuelle d'une sauvegarde . . . . .	231
	Intégration manuelle de l'historique . . . . .	234
	Remonter en arrière parmi les opérations . . . . .	234
	Le journal des sauvegardes . . . . .	236
	Fichier XML des paramètres . . . . .	236
	A propos de 4D Backup . . . . .	238
	Compatibilité . . . . .	238
	Fonctions de 4D Backup non maintenues . . . . .	238
	Avantages du nouveau module . . . . .	239
<b>Chapitre 6</b>	<b>Serveur Web . . . . .</b>	<b>241</b>
	Connexions persistantes . . . . .	241
	Quel est l'intérêt des connexions persistantes ? . . . . .	242
	Paramétrer les connexions persistantes . . . . .	242
	Modifier le numéro de port HTTPS . . . . .	243
	Prise en charge des URLs inconnus . . . . .	244
	Lancement du serveur Web sous MacOS X . . . . .	245
	Tester le serveur Web. . . . .	246
<b>Chapitre 7</b>	<b>Langage . . . . .</b>	<b>249</b>
	Formules . . . . .	251
	FIXER METHODES AUTORISEES . . . . .	251
	LIRE METHODES AUTORISEES . . . . .	252
	EDITER FORMULE . . . . .	253

---

Formulaires utilisateurs . . . . .	254
Principes de fonctionnement . . . . .	254
Formulaires utilisateurs et bibliothèques d'objets . . . . .	255
Stockage et gestion des formulaires utilisateurs. . . . .	256
MODIFIER FORMULAIRE . . . . .	256
CREER FORMULAIRE UTILISATEUR . . . . .	258
LISTE FORMULAIRES UTILISATEURS . . . . .	259
SUPPRIMER FORMULAIRE UTILISATEUR . . . . .	259
Codes d'erreurs des formulaires utilisateurs. . . . .	260
Listes hiérarchiques . . . . .	260
SELECTIONNER ELEMENTS PAR POSITION (SELECTIONNER ELEMENT) . . . . .	261
SELECTIONNER ELEMENTS PAR REFERENCE . . . . .	262
CHANGER PROPRIETES LISTE . . . . .	263
LIRE PROPRIETES LISTE . . . . .	264
CHANGER PROPRIETES ELEMENT . . . . .	265
LIRE PROPRIETES ELEMENT . . . . .	266
CHANGER ELEMENT . . . . .	267
INFORMATION ELEMENT . . . . .	267
Element parent . . . . .	268
Elements selectionnes (Element selectionne) . . . . .	268
Nombre elements . . . . .	270
Utilisateurs et groupes . . . . .	271
Nouvelles commandes . . . . .	271
ECRIRE ACCES PLUGIN . . . . .	271
Lire acces plugin . . . . .	272
LIRE LISTE PLUGIN . . . . .	273
UTILISATEURS VERS BLOB . . . . .	273
BLOB VERS UTILISATEURS . . . . .	274
Lire utilisateur par défaut . . . . .	275
Commandes modifiées . . . . .	276
CHANGER UTILISATEUR COURANT (CHANGER UTILISATEUR) . . . . .	276
LIRE PROPRIETES UTILISATEUR . . . . .	278
Ecrire proprietes utilisateur . . . . .	278
Proposer une boîte de dialogue de gestion d'accès personnalisée . . . . .	279

Fenêtres . . . . .	280
Nouvelles commandes . . . . .	280
Fenetre formulaire courant . . . . .	280
REDIMENSIONNER FENETRE FORMULAIRE . . . . .	281
Commande modifiée. . . . .	282
CHANGER COORDONNEES FENETRE . . . . .	282
Bords pousseurs . . . . .	283
Type Fenêtre pop up . . . . .	286
Type Fenêtre feuille . . . . .	287
Formulaires . . . . .	289
FIXER TAILLE FORMULAIRE . . . . .	289
FIXER REDIMENSIONNEMENT FORMULAIRE HORIZONTAL . . . . .	291
FIXER REDIMENSIONNEMENT FORMULAIRE VERTICAL . . . . .	292
LIRE OBJETS FORMULAIRE . . . . .	292
Commandes modifiées. . . . .	294
FORMULAIRE ENTREE . . . . .	294
FORMULAIRE SORTIE . . . . .	295
Liens . . . . .	295
LIRE LIEN CHAMP . . . . .	296
FIXER LIEN CHAMP . . . . .	299
LIRE LIENS AUTOMATIQUES . . . . .	301
Evénements formulaires . . . . .	301
Evénements formulaire liés aux survols . . . . .	302
Evénements formulaire liés aux boutons 3D . . . . .	303
Evénements formulaire liés aux list box . . . . .	304
Evénements formulaire liés aux enregistrements en liste .	306
Evénements formulaire liés aux listes hiérarchiques . . .	307
Interface utilisateur. . . . .	308
LIRE TITRES TABLES . . . . .	308
LIRE TITRES CHAMPS . . . . .	309
DEFILER LIGNES . . . . .	309
FIXER TITRES CHAMPS / FIXER TITRES TABLES . . . . .	310
Pop up menu . . . . .	311
Objet focus . . . . .	311
Gestion de la saisie . . . . .	312
EDITER ELEMENT . . . . .	312
Impressions. . . . .	314
Imprimer sur le serveur . . . . .	314
FIXER OPTION IMPRESSION / LIRE OPTION IMPRESSION (thème "Impressions") . . . . .	315
Messages . . . . .	316
AFFICHER NOTIFICATION . . . . .	316



---

Saisie . . . . .	317
DIALOGUE . . . . .	317
Documents système . . . . .	317
Selectionner document . . . . .	317
Propriétés des objets . . . . .	321
Lire formatage . . . . .	321
CHOIX SAISSISSABLE . . . . .	323
CHOIX FORMATAGE . . . . .	324
FIXER COULEURS RVB . . . . .	326
CHOIX COULEUR . . . . .	327
List box . . . . .	328
Principes de programmation des objets de type List box . . . . .	329
INSERER COLONNE LISTBOX . . . . .	334
SUPPRIMER COLONNE LISTBOX . . . . .	335
Lire nombre colonnes listbox . . . . .	336
TRIER COLONNES LISTBOX . . . . .	336
FIXER LARGEUR COLONNE LISTBOX . . . . .	337
Lire largeur colonne listbox . . . . .	338
NUMERO COLONNE LISTBOX DEPLACÉE . . . . .	339
SELECTIONNER LIGNE LISTBOX . . . . .	339
INSERER LIGNE LISTBOX . . . . .	340
SUPPRIMER LIGNE LISTBOX . . . . .	341
Lire nombre lignes listbox . . . . .	341
FIXER HAUTEUR LIGNES LISTBOX . . . . .	342
Lire hauteur lignes listbox . . . . .	342
NUMERO LIGNE LISTBOX DEPLACÉE . . . . .	343
LIRE TABLEAUX LISTBOX . . . . .	344
Lire information listbox . . . . .	345
MONTRER GRILLE LISTBOX . . . . .	346
FIXER COULEUR GRILLE LISTBOX . . . . .	347
MONTRER BARRES DEFILEMENT LISTBOX . . . . .	348
Environnement 4D . . . . .	348
OUVRIR PREFERENCES 4D . . . . .	348
Environnement système. . . . .	352
FIXER VARIABLE ENVIRONNEMENT . . . . .	352
LANCER PROCESS EXTERNE . . . . .	353
Sélections . . . . .	355
LIRE ENREGISTREMENTS MARQUES . . . . .	355
Numero de ligne affichee . . . . .	356
MARQUER ENREGISTREMENTS . . . . .	357
MODIFIER SELECTION, VISUALISER SELECTION . . . . .	358
ALLER DANS SELECTION . . . . .	359

Serveur Web . . . . .	360
TRAITER BALISES HTML . . . . .	360
LIRE CORPS HTTP . . . . .	362
ENVOYER DONNEES HTTP . . . . .	363
LIRE VARIABLES FORMULAIRE WEB . . . . .	366
Prise en charge des CGI . . . . .	366
FIXER EXECUTABLE CGI . . . . .	366
Tableaux . . . . .	368
Compter dans tableau . . . . .	368
AJOUTER A TABLEAU . . . . .	368
Communications . . . . .	369
LIRE CORRESPONDANCE PORT SERIE . . . . .	369
Sauvegarde . . . . .	370
SAUVEGARDER . . . . .	370
RESTITUER . . . . .	371
LIRE INFORMATION SAUVEGARDE . . . . .	371
LIRE INFORMATION RESTITUTION . . . . .	372
Méthode base Sur démarrage sauvegarde . . . . .	373
Méthode base Sur arrêt sauvegarde . . . . .	373
Codes d'erreurs de Sauvegarde . . . . .	374
XML . . . . .	374
Changement de nom des commandes 4D 2003. . . . .	375
Prise en charge du standard SAX. . . . .	375
SAX ECRIRE OPTIONS XML . . . . .	376
SAX AJOUTER DOCTYPE XML . . . . .	377
SAX AJOUTER COMMENTAIRE XML . . . . .	378
SAX OUVRIR ELEMENT XML . . . . .	378
SAX OUVRIR ELEMENT XML TABLEAUX . . . . .	379
SAX FERMER ELEMENT XML . . . . .	380
SAX AJOUTER VALEUR ELEMENT XML . . . . .	381
SAX AJOUTER CDATA XML . . . . .	381
SAX AJOUTER INSTRUCTION DE TRAITEMENT . . . . .	383
SAX Lire noeud XML . . . . .	383
SAX LIRE VALEURS DOCUMENT XML . . . . .	385
SAX LIRE COMMENTAIRE XML . . . . .	385
SAX LIRE INSTRUCTION DE TRAITEMENT XML . . . . .	386
SAX LIRE ELEMENT XML . . . . .	386
SAX LIRE VALEUR ELEMENT XML . . . . .	387
SAX LIRE CDATA XML . . . . .	388
SAX LIRE ENTITE XML . . . . .	389
Ecriture de XML via les commandes DOM . . . . .	389
DOM ECRIRE OPTIONS XML . . . . .	391
DOM Creer ref XML . . . . .	392
DOM ECRIRE ATTRIBUT XML . . . . .	393
DOM Creer element XML . . . . .	394

DOM ECRIRE NOM ELEMENT XML	396
DOM ECRIRE VALEUR ELEMENT XML	397
DOM SUPPRIMER ELEMENT XML	398
DOM EXPORTER VERS FICHER	398
DOM EXPORTER VERS VARIABLE	399
DOM Chercher element XML	399
Commande modifiée	400
DOM LIRE VALEUR ELEMENT XML	400
Codes d'erreurs XML	401
Prise en charge des transformations XSL	401
APPLIQUER TRANSFORMATION XSLT	403
FIXER PARAMETRE XSLT	404
LIRE ERREUR XSLT	405
Web Services (Client)	406
FIXER OPTION WEB SERVICE	406
Source de données externes	408
ODBC LOGIN	409
ODBC LOGOUT	410
ODBC FIXER OPTION	411
ODBC LIRE OPTION	412
ODBC EXECUTER	412
ODBC Fin de selection	414
ODBC CHARGER ENREGISTREMENT	415
ODBC ANNULER CHARGEMENT	415
ODBC FIXER PARAMETRE	416
ODBC LIRE DERNIERE ERREUR	419
ODBC IMPORTER	419
ODBC EXPORTER	421
Outils	423
GENERER APPLICATION	423
ENCODER	424
DECODER	425
CORRECTION ORTHOGRAPHIQUE	425
CHANGER DICTIONNAIRE	426

<b>Index</b>	<b>427</b>
--------------	------------



# 1

# Introduction

Bienvenue dans 4<sup>e</sup> Dimension 2004, la nouvelle version de l'environnement de développement 4<sup>e</sup> Dimension et 4D Server.

## Présentation

4<sup>e</sup> Dimension 2004 apporte de multiples nouveautés et optimisations concernant la plupart des fonctions du programme. Ces nouveautés se déclinent autour de cinq axes principaux :

Tout d'abord, une plus grande **intégration avec les systèmes d'exploitation de dernière génération**, en particulier Windows XP et MacOS X. Cette intégration accrue se traduit notamment par une meilleure prise en charge des spécificités d'interface, de la gestion mémoire, le support du format Mach-O sous MacOS X, etc. En contrepartie, la gamme 4<sup>e</sup> Dimension 2004 n'est plus compatible avec les systèmes d'exploitation MacOS 9.x d'Apple, Windows 98 S.E. et Windows Millenium.

Ensuite, 4<sup>e</sup> Dimension 2004 poursuit ses efforts d'**ouverture vers les technologies les plus reconnues**. Outre le renforcement de la prise en charge des Web Services et des CGI, 4<sup>e</sup> Dimension 2004 propose de nombreuses possibilités d'ouverture vers les applications et les systèmes externes, notamment via de nouvelles commandes XML, la prise en charge de XSL, l'importation et l'exportation de données au format ODBC et l'ajout de commandes ODBC de haut niveau.

L'**atelier de développement** a également subi de nombreuses modifications : barre d'outils plus fonctionnelle, modernisation de l'éditeur de formulaires (zoom, magnétisme, vues...), nouvelle boîte à outils, possibilité de définir des groupes d'objets dans l'Explorateur, nouveaux événements formulaire et une centaine de nouvelles commandes du langage sont quelques-unes des nombreuses améliorations apportées à l'environnement de travail des développeurs 4<sup>e</sup> Dimension.

4<sup>e</sup> Dimension 2004 propose plusieurs nouveaux outils permettant d'enrichir l'**interface utilisateur** des applications finales, comme les nouveaux objets de formulaire "list box" et boutons "bevel", l'amélioration de la gestion des listes d'enregistrements (sous-formulaires, formulaires sortie...) ou la possibilité pour l'utilisateur de modifier des formulaires à sa convenance et d'accéder à une bibliothèque d'objets.

Enfin, de nouveaux **outils de déploiement** ont été mis à la disposition des développeurs, notamment une solution intégrée de sauvegarde souple et efficace (gestion des sauvegardes, des restitutions et du fichier d'historique), un générateur programmable d'applications client/serveur entièrement personnalisées et un mécanisme de mise à jour automatique des parties clientes.

## Contenu de ce manuel

Ce manuel détaille toutes les nouveautés et modifications introduites dans la version 2004 de 4<sup>e</sup> Dimension et de 4D Server. Il se compose des chapitres suivants :

- **Architecture des applications** Ce chapitre décrit les modifications concernant l'architecture interne des fichiers des applications 4<sup>e</sup> Dimension 2004 et des plug-ins.
- **Mode Structure** Ce chapitre liste les nouveautés et modifications concernant le mode Structure : nouvelles Préférences générales, regroupement sous forme de "boîte à outils" et modernisation des éditeurs, possibilité de "cloner" une table ODBC, modification de l'éditeur de formulaires, nouveaux objets de formulaires, prise en charge étendue des Web Services, et plus encore.

- **Mode Utilisation** Ce chapitre décrit les nouveautés relatives au mode Utilisation de 4<sup>e</sup> Dimension : nouvel éditeur de formules, importation et exportation de données au format ODBC et modification de l'éditeur d'états rapides.
- **Module de sauvegarde intégré** Ce chapitre décrit le fonctionnement du nouveau module de sauvegarde intégré, permettant la sauvegarde et la restitution de vos données, ainsi que la gestion du fichier d'historique.
- **Serveur Web** Ce chapitre traite des modifications effectuées sur le serveur Web de 4<sup>e</sup> Dimension 2004, notamment l'amélioration de l'usage de CGI et la prise en charge du protocole HTTP 1.1.
- **Langage** Ce chapitre détaille les nouveautés et modifications apportées au langage de 4<sup>e</sup> Dimension 2004 : nouveaux thèmes de commandes, nouvelles commandes, nouvelles méthodes base, commandes modifiées...

Des modifications ont également été apportées aux plug-ins de la gamme 4<sup>e</sup> Dimension 2004, notamment 4D Pack, 4D Write, 4D View et 4D Draw. Ces nouveautés sont décrites dans le manuel *Mise à jour des plug-ins*.

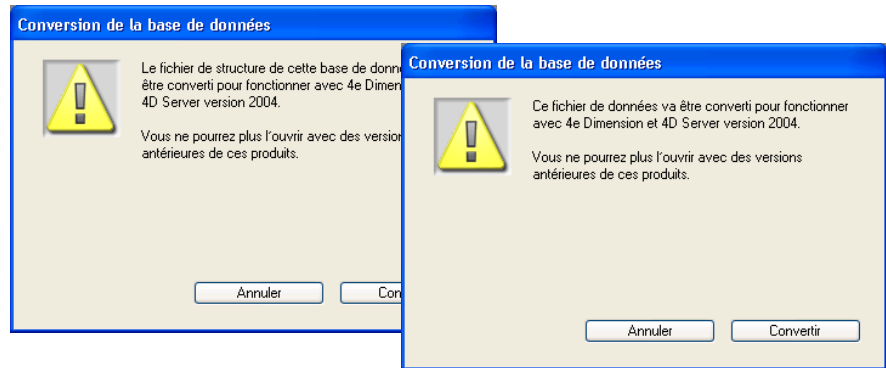
4<sup>e</sup> Dimension version 2004 s'accompagne d'un nouveau plug-in pour ODBC. Ce plug-in contient des commandes de bas niveau, permettant de mettre en place tout type de connexion avec des applications compatibles ODBC. Ce plug-in fait l'objet d'une documentation séparée.

## Compatibilité avec les versions précédentes

Les bases de données créées avec des versions 6.7.x, 6.8.x ou 2003 de 4<sup>e</sup> Dimension ou de 4D Server sont entièrement compatibles avec les applications 4<sup>e</sup> Dimension 2004 (fichier de structure et fichier de données).

Toutefois, les bases sont converties à l'ouverture par 4<sup>e</sup> Dimension 2004 (fichier de structure et fichier de données). Vous ne pourrez plus les ouvrir avec une version précédente de 4<sup>e</sup> Dimension.

Lorsque vous ouvrez avec 4<sup>e</sup> Dimension 2004 une base créée avec une version précédente, deux boîtes de dialogue successives vous informent que le fichier de structure et le fichier de données vont être convertis :



## Configuration minimale

Le renforcement de l'intégration de 4<sup>e</sup> Dimension 2004 avec les systèmes d'exploitation de dernière génération a conduit 4D à cesser de prendre en charge les versions les plus anciennes de ces systèmes. Par conséquent, à compter de la version 2004 de 4<sup>e</sup> Dimension :

- **Macintosh** : le système d'exploitation MacOS 9.x n'est plus pris en charge. La version minimale sur Macintosh est désormais la version 10.1.
- **Windows** : les systèmes d'exploitation Windows 98 S.E. et Windows Millenium ne sont plus pris en charge. La version minimale sous Windows est désormais Windows 2000.

Les applications de la gamme 4<sup>e</sup> Dimension version 2004 requièrent au minimum les configurations suivantes :

	Windows	MacOS
Ordinateur	Compatible PC avec microprocesseur Pentium II	Macintosh iMac
Système d'exploitation	Windows 2000, Windows XP	MacOS version 10.2.8 ou ultérieure
Mémoire minimale	128 Mo	128 Mo
Mémoire conseillée	256 Mo	256 Mo
Résolution écran	800*600 pixels	



# 2

## Architecture des applications 4D

### 4<sup>e</sup> Dimension au format Mach-O (MacOS)

Les versions MacOS des applications moteur 4D 2004 (4<sup>e</sup> Dimension, 4D Server, 4D Client et 4D Runtime) et des principaux plug-ins 4D ont été portées au format *Mach-O*.

#### Qu'est-ce que le format Mach-O ?

Mach-O (Mach Object-file format) est le format natif des applications exécutables sous MacOS X. Les applications fournies dans ce format sont donc mieux intégrées au système, tant au niveau du contrôle des privilèges d'accès que du fonctionnement en tant que service d'arrière-plan. Les performances générales sont en outre améliorées sous MacOS X.

Le format Mach-O permet notamment aux applications 4D :

- de lancer le serveur Web sur le port TCP 80 (port standard) lorsque l'utilisateur courant n'est pas l'utilisateur *Root*,
- d'enregistrer un moteur 4D (4<sup>e</sup> Dimension, 4D Server, 4D Client, 4D Runtime) en tant que service et d'ouvrir automatiquement une base de données.

#### Mach-O et MacOS 9

Le format Mach-O n'est pas compatible avec MacOS 9.

Pour cette raison, les progiciels (ou *packages*) 4<sup>e</sup> Dimension 2004 ne peuvent pas fonctionner sur la plate-forme MacOS 9 ni en mode Classic sous MacOS X.

## Compatibilité des plug-ins

Les plug-ins d'architecture non *carbon* (ancienne génération) ne sont plus compatibles avec la gamme 4<sup>e</sup> Dimension 2004 pour MacOS. Pour pouvoir être chargés par 4<sup>e</sup> Dimension, les plug-ins doivent être *carbon* et au format Mach-O ou *CFM*.

---

*Note* CFM (Code Fragment Manager) est l'“ancienne” librairie des plug-ins. Lorsqu'ils sont *carbon*, ces plug-ins restent compatibles MacOS X.

---

La plupart des plug-ins 4D passent au format Mach-O :

- 4D View
- 4D Write
- 4D Draw
- 4D for OCI
- 4D ODBC Pro
- 4D Open for 4D
- 4D Pack
- 4D Internet Commands

Les plug-ins 4D suivants ne sont plus pris en charge en version 2004 :

- 4D Calc
- 4D List
- 4D ODBC

Les plug-ins 4D 2003 et antérieurs ne sont pas compatibles avec 4<sup>e</sup> Dimension 2004. Seuls les plug-ins 4D en version 2004 peuvent être utilisés avec 4<sup>e</sup> Dimension 2004. Cette restriction ne concerne pas les plug-ins commercialisés par des sociétés tierces.

## Nouvelle architecture des plug-ins

L'architecture des plug-ins au sein de 4<sup>e</sup> Dimension 2004 a été modifiée, tant au niveau de l'emplacement des fichiers des plug-ins que de la structure interne des plug-ins.

### Emplacements des plug-ins

Dans 4<sup>e</sup> Dimension 2004, les plug-ins peuvent être placés à deux endroits :

- au même niveau que la structure de la base,
- au même niveau que l'application 4<sup>e</sup> Dimension.

---

*Note* Les plug-ins placés au niveau du dossier 4D actif ne sont donc plus chargés par 4<sup>e</sup> Dimension 2004.  
A noter que les préférences des plug-ins sont toujours stockées dans le dossier 4D actif.

---

De plus, il est possible d'utiliser deux types de dossiers pour stocker les plug-ins :

- le nouveau dossier **PlugIns**,
- un dossier **Mac4DX / Win4DX** (au niveau de la structure de la base uniquement).

Les paragraphes suivants détaillent les principes d'installation des plug-ins.

### Plug-ins placés au niveau de l'application 4D

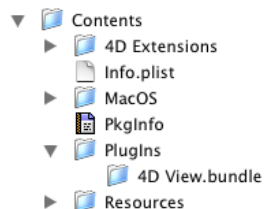
Les plug-ins placés au même niveau que l'application 4D sont disponibles dans toutes les bases de données ouvertes par cette application.

A cet emplacement, seul le dossier **PlugIns** est pris en charge. Il n'est pas possible de placer un dossier Win4DX/Mac4DX à cet endroit.

Les plug-ins MacOS copiés dans ce dossier doivent impérativement être fournis sous forme de *progiciels* (packages), quel que soit leur format (Mach-O ou CFM). Les plug-ins n'étant pas sous forme de progiciels ne peuvent être placés que dans les dossiers Mac4DX/Win4DX.

Sous Windows, les plug-ins copiés dans ce dossier doivent reproduire la structure des progiciels Macintosh (cf. [paragraphe "Architecture interne des plug-ins"](#), page 21).

- Sous MacOS, le dossier **PlugIns** est placé dans le dossier **Contents** du progiciel :

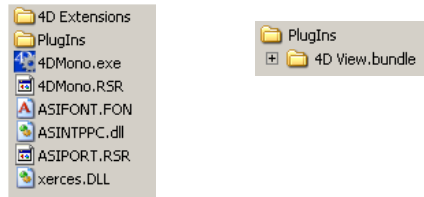



---

*Note* Rappelons que, pour visualiser le contenu d'un progiciel sous MacOS X, vous devez cliquer sur son icône tout en maintenant la touche **Control** enfoncée, puis choisir la commande **Afficher le contenu du paquet** dans le menu contextuel.

---

- Sous Windows, le dossier **PlugIns** est placé au même niveau que l'application 4D :

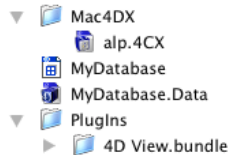


### Plug-ins placés au niveau de la structure de la base

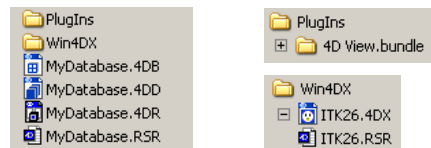
Les plug-ins placés au niveau de la structure de la base de données sont disponibles dans cette base de données uniquement.

A cet emplacement, vous pouvez utiliser soit un dossier **PlugIns**, soit un dossier **Win4DX** ou **Mac4DX**.

- Sous MacOS, le dossier **PlugIns** doit contenir les plug-ins fournis sous forme de progiciels et le dossier **Mac4DX** doit contenir les autres plug-ins (architecture standard).  
Il est cependant possible de placer un plug-in sous forme de progiciel dans le dossier **Mac4DX**.



- Sous Windows, le dossier **PlugIns** doit contenir les plug-ins respectant la structure des progiciels Macintosh (cf. [paragraphe "Architecture interne des plug-ins", page 21](#)), et le dossier **Win4DX** doit contenir les autres plug-ins (architecture standard).  
Il est cependant possible de placer un plug-in de structure type progiciel dans le dossier **Win4DX**.



- Priorité de chargement** Si un même plug-in existe à plus d'un emplacement, 4<sup>e</sup> Dimension applique les règles suivantes :
- si deux instances d'un même plug-in sont situées à des niveaux différents (structure de la base et application), 4<sup>e</sup> Dimension charge uniquement le plug-in situé à côté de la structure.
  - si deux instances d'un même plug-in sont situées aux deux emplacements (Win4DX/Mac4DX et PlugIns) mais au même niveau (structure de la base ou application), 4<sup>e</sup> Dimension affiche un message d'erreur et la base ne peut être lancée.

---

*Note* Dans le cas d'une application compilée et fusionnée avec *4D Runtime Volume License* (cf. [Nouveaux noms pour les applications de déploiement](#)), la présence de plusieurs instances d'un même plug-in empêchera l'ouverture de l'application.

---

**Architecture interne des plug-ins** Les plug-ins "nouvelle génération" doivent respecter l'architecture type progiciel sous MacOS pour pouvoir être exploités dans le dossier **PlugIns**. Cette architecture est également préconisée sous Windows.

---

*Note* Les plug-ins ne respectant pas cette architecture doivent être placés dans les dossiers **Win4DX / Mac4DX** à côté de la structure de la base.

---

Le nom du dossier du plug-in doit comporter l'extension **".bundle"**. Chaque bundle de plug-in ne comporte qu'une seule langue.

**Version monoposte** Avec 4<sup>e</sup> Dimension version monoposte, l'architecture interne des plug-ins doit être la suivante :

- Sous MacOS :
  - MonPlugIn.bundle/
    - Contents/
      - Info.plist
      - MacOS/
        - MonPlugIn (code Mach-O ou CFM)
        - PkgInfo
        - Resources/
          - MonPlugIn.rsrc (ressources)

- Sous Windows :  
MonPlugIn.bundle/  
  Contents/  
    Windows/  
      MonPlugIn.4DX  
      MonPlugIn.RSR

**Version client-serveur**

Dans le cadre d'une configuration client-serveur multi-plate-forme, il suffit de placer les dossiers Windows et MacOS (ne pas oublier le dossier Resources) dans le même dossier .bundle. Typiquement, un plug-in multi-plate-forme contiendra l'architecture suivante :

- MonPlugIn.bundle/  
  Contents/  
    Info.plist  
    MacOS/  
      MonPlugIn (code Mach-O ou CFM)  
    PkgInfo  
    Resources/  
      MonPlugIn.rsrc (ressources)  
    Windows/  
      MonPlugIn.4DX  
      MonPlugIn.RSR

---

*Note* Les fichiers *Info.plist* et *PkgInfo* sont utiles uniquement pour les plug-ins exécutés sous MacOS. Par conséquent, si vous configurez un dossier bundle multi-plate-forme sur 4D Server Windows, ces fichiers sont facultatifs.

---

**Mécanismes de téléchargement des plug-ins en client-serveur**

Lors de la connexion d'un poste client (4D Client) à un 4D Server, les plug-ins installés sur le serveur sont rapatriés en local sur le poste client. En fonction de la configuration du serveur, les plug-ins téléchargés sont placés dans un dossier "PlugIns" et/ou "Mac4DX/Win4DX" sur le poste client, comme décrit dans les tableaux suivants :

- Si les dossiers **PlugIns** et/ou **Mac4DX/Win4DX** sont installés sur le poste serveur au même niveau que la structure de la base :

Systeme	Emplacement des dossiers téléchargés sur le poste client
Mac OS X	{Disque}:Library:Support aux applications:4D:NomDeLaBase:PlugIns (et/ou Mac4DX ou Win4DX)

Windows 2000	{Disque}:\Documents and Settings\Nom de l'utilisateur\Application Data\4D\Nom de la Base\PlugIns (et/ou Mac4DX ou Win4DX)
Windows XP	

- Si le dossier **PlugIns** est installé sur le poste serveur au même niveau que l'application 4D Server :

Systeme	Emplacement des dossiers téléchargés sur le poste client
Mac OS X	{Disque}:Library:Support aux applications:4D:NomDeLaBase:4D:PlugIns
Windows 2000	{Disque}:\Documents and Settings\Nom de l'utilisateur\Application Data\4D\Nom de la Base\4D\PlugIns
Windows XP	

*Note* Avec 4D Server 2004, la hiérarchie des éléments (dossiers et fichiers) placés dans le dossier de téléchargement est conservée et reconstituée sur les postes clients.

Seuls les éléments correspondants à la plate-forme du poste client sont téléchargés.

*Note* Des dossiers supplémentaires nommés Cache, Server et Database sont également créés sur les postes clients pour des besoins de fonctionnement internes à 4D Server.

## Command Line Interface (CLI) sous MacOS X

Vous pouvez utiliser une ligne de commande dans le Terminal de MacOS X afin de piloter vos applications 4<sup>e</sup> Dimension (4<sup>e</sup> Dimension, 4D Server et 4D Client). Cette fonction permet de quitter ou d'ouvrir une base de données à distance, ce qui est particulièrement utile pour les administrateurs de serveurs Web.

### Principes d'utilisation

Pour pouvoir exécuter la plupart des commandes relatives aux applications 4<sup>e</sup> Dimension à l'aide du Terminal de MacOS X, vous devez au préalable accéder au dossier dans lequel se trouve l'application, à l'intérieur du progiciel (chemin Contents/MacOS). Par exemple, si le progiciel 4<sup>e</sup> Dimension se trouve dans le dossier *MonDossier*, vous devez écrire la ligne de commande ainsi :

```
/MonDossier/4e\ Dimension.app/Contents/MacOS/4e\ Dimension
```

*Note* Vous pouvez également insérer la ligne de commande via un glisser-déposer de l'icône de l'application dans le Terminal.

Il est préférable de passer le caractère & à la fin de la ligne de commande afin que l'application soit exécutée en tâche de fond dans un process séparé. Par exemple :

```
/MonDossier/4e\ Dimension.app/Contents/MacOS/4e\ Dimension &
```

### Commandes et arguments

Voici la description des lignes de commandes et de leurs arguments pris en charge par les applications 4<sup>e</sup> Dimension :

#### ■ Lancer l'application

*Syntaxe* : Chemin/NomApplication

*Exemple* :

```
4e\ Dimension.app/Contents/MacOS/4e\ Dimension &
```

L'action de cette commande est identique à un double-clic sur l'application 4<sup>e</sup> Dimension. L'application s'exécute et la boîte de dialogue de sélection de base de données s'affiche.

#### ■ Lancer l'application avec un fichier de structure

*Syntaxe* : Chemin/NomApplication CheminStructure

*Exemple* :

```
4e\ Dimension.app/Contents/MacOS/4e\ Dimension /Users/Roger/Bases  
/MaBase/MaBase.4DB
```

Cette commande lance l'application et ouvre le fichier de structure désigné avec le fichier de données courant. Aucune boîte de dialogue n'apparaît.

#### ■ Lancer l'application avec un fichier de structure et un fichier de données

*Syntaxe* : Chemin/NomApplication -d CheminDonnées CheminStructure

*Exemple* :

```
4e\ Dimension.app/Contents/MacOS/4e\ Dimension -d /Users/Roger  
/Bases/MaBase/MaBase.4DD /Users/Roger/Bases/MaBase/MaBase.4DB
```

Cette commande lance l'application et ouvre le fichier de structure désigné avec le fichier de données désigné. Aucune boîte de dialogue n'apparaît.



### ■ Quitter l'application

*Syntaxe* : kill -s INT process\_ID

*Exemple* :

Kill -s INT 323

L'action de cette commande est identique à la sélection de la commande **Quitter** dans le menu **Fichier**. Les process sont refermés un à un, le cache est enregistré et l'application quitte.

---

*Note* Le numéro de process de l'application 4<sup>e</sup> Dimension (*process\_ID*) peut être obtenu à l'aide de commandes telles que **ps** ou **top**.

---

### ■ Aide

*Syntaxe* : Chemin/NomApplication -h

*Exemple* :

4e\ Dimension.app/Contents/MacOS/4e\ Dimension -h

Cette commande affiche un résumé des commandes utilisables avec 4<sup>e</sup> Dimension dans le Terminal de MacOS X.

### ■ Lancer l'application dans une session utilisateur différente de la session courante

*Syntaxe* : Chemin/NomApplication -u NomUtilisateur -g NomGroupe

*Exemple* :

4e\ Dimension.app/Contents/MacOS/4e\ Dimension -u john -g compta

Cette commande force le lancement de l'application au sein de la session utilisateur spécifiée (par défaut, la session utilisateur courante est utilisée). Ce fonctionnement permet de définir la session utilisée après le lancement du serveur Web via une session root temporaire (cf. [paragraphe "Lancement du serveur Web sous MacOS X", page 245](#)).

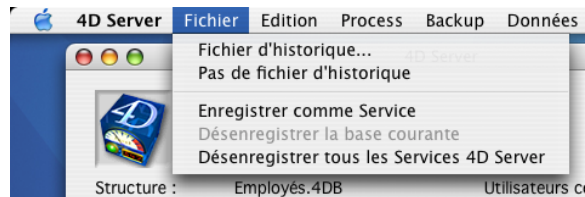
## Enregistrer 4D Server comme service sous MacOS X

L'enregistrement de 4D Server 2004 en tant que service est désormais simplifié sous MacOS X.

*Note* Le mécanisme d'enregistrement de 4D Server comme service sous Windows n'a pas été modifié.

Une application 4D Server enregistrée comme service est automatiquement lancée au démarrage de la machine (avec la base courante), avant même l'ouverture d'une session utilisateur. Ce principe permet de garantir la disponibilité d'une base 4D Server, même en cas d'incident nécessitant le redémarrage de la machine. La maintenance peut être effectuée à distance.

Comme sous Windows, l'enregistrement et le désenregistrement des bases 4D Server comme services sous MacOS X s'effectuent via des commandes spécifiques dans le menu **Fichier** du poste serveur :



- **Enregistrer comme service** : enregistre la base courante comme service. Au prochain démarrage de la machine, 4D Server sera automatiquement lancé et la base courante ouverte.
- **Désenregistrer la base courante** : désenregistre la base courante comme service. Cette commande est grisée si la base n'est pas enregistrée comme service.
- **Désenregistrer tous les services 4D Server** : désenregistre tous les services 4D Server déclarés sur le poste. Cette commande est grisée si aucun service 4D Server n'est actif.

## Fichier d'aide syntaxique

Le fichier d'aide syntaxique a été modifié dans 4<sup>e</sup> Dimension 2004. Ce fichier permet d'afficher et de pré-saisir la syntaxe des commandes dans l'éditeur de méthodes, ainsi que d'afficher une description des commandes dans l'Explorateur sous Windows :



Dans les versions précédentes de 4<sup>e</sup> Dimension, ce fichier était nommé 4D Help (MacOS) et 4D Help.rsr (Windows) et devait être placé dans le dossier 4D actif.

Afin de préciser son usage et de simplifier ses mises à jour, ce fichier est désormais nommé **4D Syntax.rsr** sur les deux plates-formes et doit être placé dans le dossier **4D Extensions** de l'application 4D.



# 3

## Mode Utilisation

Plusieurs nouveautés de 4<sup>e</sup> Dimension 2004 concernent le mode Utilisation :

- Nouvel éditeur de formulaires utilisateurs
- Modifications de l'éditeur de formules
- Importation et exportation de ou vers une source ODBC
- Editeur d'états rapides : génération automatique du code et ajout de la fonction Ecart type.
- Diverses modifications concernant l'interface de 4<sup>e</sup> Dimension et le mode Utilisation sont traitées dans le chapitre "Mode Structure", [paragraphe "Interface", page 71](#).

### Editeur de formulaires utilisateurs

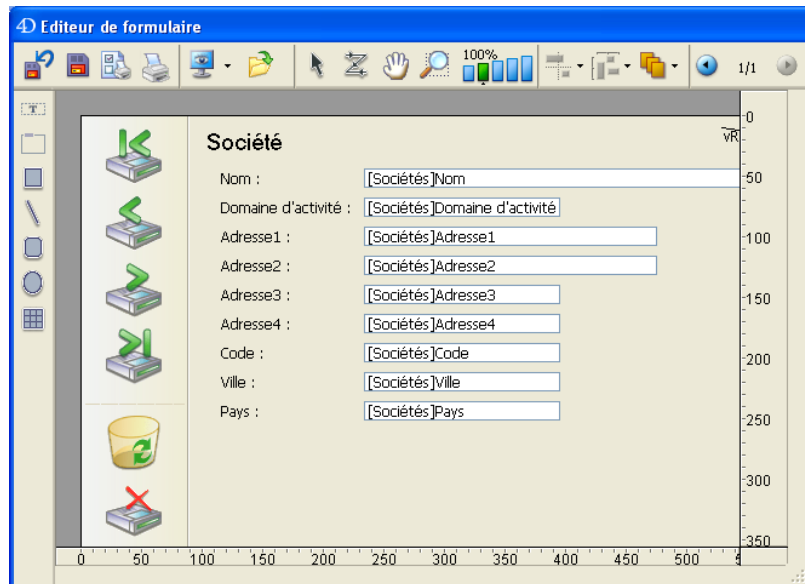
Dans 4<sup>e</sup> Dimension 2004, le développeur peut permettre aux utilisateurs de modifier les formulaires de la base afin, par exemple, de créer des états personnalisés. Une fois enregistré, un "formulaire utilisateur" est utilisable comme n'importe quel formulaire de 4<sup>e</sup> Dimension.

Les mécanismes de mise en place de ce système sont décrits dans le chapitre Langage, [paragraphe "Formulaires utilisateurs", page 254](#). Le présent paragraphe décrit le fonctionnement de l'éditeur de formulaires utilisateurs, dans lequel l'utilisateur peut personnaliser le formulaire.

#### Description de l'éditeur

L'éditeur de formulaires utilisateurs s'affiche lorsque la commande **MODIFIER FORMULAIRE** est exécutée, si la propriété **Modifiable par l'utilisateur** a bien été cochée pour le formulaire passé en paramètre.

Il apparaît ainsi :



Cet éditeur reprend en partie l'interface de l'éditeur de formulaires standard de 4<sup>e</sup> Dimension, notamment le principe des deux barres intégrées (outils et objets). Il comporte également des fonctions spécifiques.

Lorsque cet éditeur est affiché, seul le menu **Edition** de la barre de menus est accessible. Il est cependant possible d'afficher une Liste des propriétés simplifiée ainsi qu'un menu contextuel.

Les mécanismes de sélection et de déplacement d'objets (lorsqu'ils sont autorisés) sont identiques à ceux de l'éditeur de formulaires standard de 4<sup>e</sup> Dimension.

### Barre d'outils

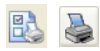
La barre d'outils comporte les éléments suivants :



**Version originale** : retourne au formulaire d'origine tel que défini en mode Structure (supprime toutes les modifications effectuées dans le formulaire utilisateur). Lorsque vous cliquez sur ce bouton, une boîte de dialogue d'alerte vous permet d'annuler ou de confirmer l'opération.



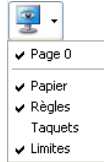
**Sauvegarde** : enregistre les modifications effectuées dans le formulaire utilisateur.



**Configuration de l'impression et Impression** : ces deux commandes permettent de paramétrer et de lancer l'impression du formulaire utilisateur.



**Affichage** : permet de sélectionner les éléments à afficher et à masquer dans l'éditeur via un pop up menu :

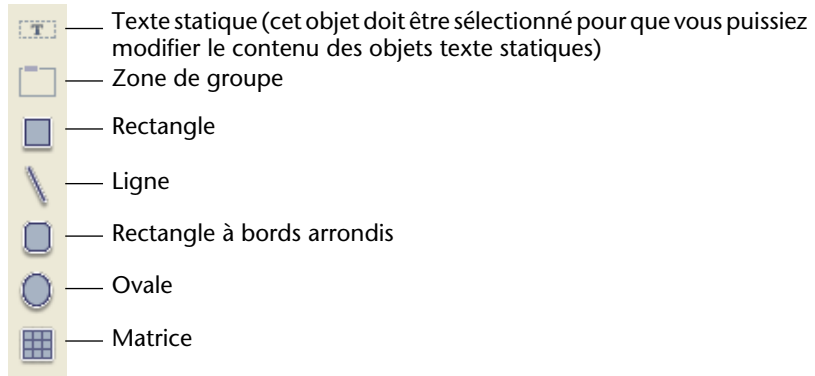


**Ouverture de la bibliothèque** : permet de charger la bibliothèque d'objets utilisable avec le formulaire le cas échéant (cf. [paragraphe "Accès aux bibliothèques d'objets"](#), page 34).

Les autres outils sont identiques à ceux de la barres d'outils de l'éditeur de formulaires standard de 4<sup>e</sup> Dimension. Pour plus d'informations, reportez-vous au [paragraphe "Barre d'outils"](#), page 116.

## Barre d'objets

La barre d'objets de l'éditeur de formulaires utilisateurs permet d'ajouter divers objets graphiques dans le formulaire :

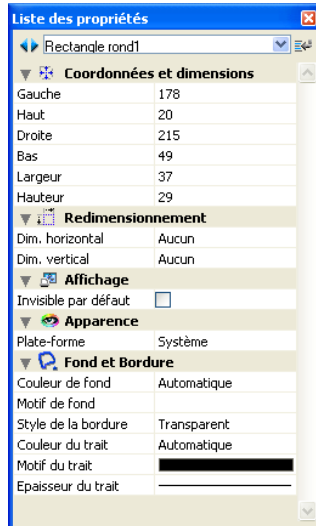


Pour créer un objet dans le formulaire, il suffit de le sélectionner dans la barre d'objets puis de le tracer dans le formulaire. Ces objets sont identiques à ceux utilisables en mode Structure.

## Visualisation et modification des objets

L'utilisateur peut visualiser et modifier les propriétés des objets présents dans le formulaire à l'aide d'une Liste des propriétés simplifiée. Pour afficher la liste, il suffit de **double-cliquer** sur un objet ou de choisir la commande **Liste des propriétés** dans le menu

contextuel de l'éditeur (clic droit sous Windows ou Control+clic sous MacOS).



Seules les propriétés modifiables par l'utilisateur sont affichées dans la Liste des propriétés.

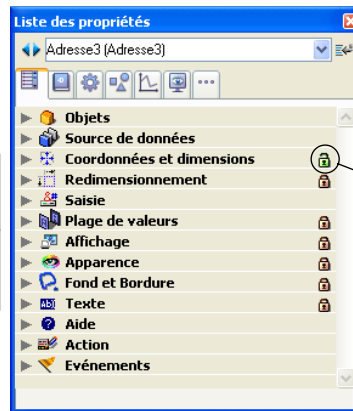
- Pour les objets graphiques ajoutés par l'utilisateur, toutes les propriétés liées au type d'objet sont disponibles.
- En ce qui concerne les objets provenant du formulaire original, deux conditions sont nécessaires pour qu'une propriété soit modifiable dans l'éditeur de formulaires utilisateurs :
  - la propriété doit appartenir à l'ensemble des propriétés pouvant être déclarées modifiables pour l'objet ou le formulaire. Par exemple, les propriétés relatives à la gestion des événements formulaire ou du glisser-déposer ne sont jamais modifiables par l'utilisateur. Dans l'éditeur de formulaires en mode Structure, les thèmes de propriétés modifiables sont repérés via une icône de cadenas dans la Liste des propriétés (cf. [paragraphe "Modifiable par l'utilisateur"](#), page 130).
  - le thème de propriétés doit avoir été explicitement déclaré comme modifiable par le développeur pour l'objet ou le formulaire. Pour cela, le développeur doit avoir cliqué sur le cadenas dans la Liste des



propriétés en mode Structure afin de le déverrouiller :

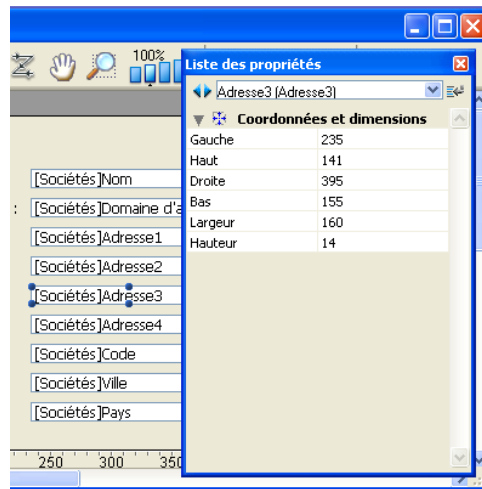
Liste des propriétés  
en mode Structure

Thèmes  
modifiables



Thème  
déverrouillé

La propriété ainsi déverrouillée devient accessible à l'utilisateur dans l'éditeur de formulaires utilisateurs :



Ce principe fonctionne également pour les autres actions possibles dans le formulaire utilisateur : par exemple, si le thème “Coordonnées et dimensions” est déverrouillé, l'utilisateur pourra déplacer l'objet à l'aide de la souris ou du clavier.

### Commandes du menu Edition

Seul le menu **Edition** est actif lorsque la fenêtre de l'éditeur de formulaires utilisateurs est affichée en mode Utilisation. Dans le cadre des interfaces personnalisées (Menus créés), le menu **Edition** est accessible si les commandes qu'il contient sont gérées via des actions standard.

Toutes les commandes d'édition (**Couper**, **Copier**, **Coller**, **Effacer**, **Tout sélectionner**) fonctionnent sans restriction avec les objets ajoutés par l'utilisateur.

En revanche, pour des raisons de cohérence, l'utilisateur n'est jamais autorisé à supprimer un objet issu du formulaire original. L'utilisation des commandes **Couper**, **Effacer** ou des touches **Suppr** ou **Ret. Arr.** provoque l'affichage d'une boîte de dialogue d'alerte informant l'utilisateur que cette action n'est pas autorisée.

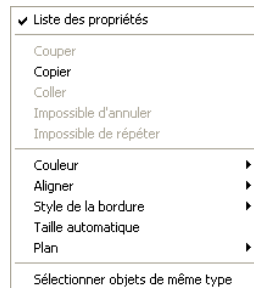
### Menu contextuel

Plusieurs commandes sont accessibles via le menu contextuel de l'éditeur de formulaires utilisateurs. Il n'est pas possible de contrôler le contenu de ce menu.

- Lorsque l'utilisateur clique en-dehors de tout objet, les commandes globales d'affichage de l'éditeur de formulaire sont accessibles : affichage de la Liste des propriétés, activation de la grille magnétique et choix des éléments à afficher et à masquer dans l'éditeur :



- Lorsque l'utilisateur clique sur un objet, des commandes graphiques sont disponibles (couleur, alignement, etc.).



L'accès à ces commandes peut toutefois être restreint pour les objets issus du formulaire original en fonction des paramètres effectués en mode Structure.

### Accès aux bibliothèques d'objets

Il est possible d'utiliser une bibliothèque d'objets avec chaque formulaire utilisateur en cours de modification.

*Note* Pour plus d'informations sur les bibliothèques d'objets, reportez-vous au paragraphe "Bibliothèques d'objets", page 175.

Le développeur désigne la bibliothèque associée au formulaire via le dernier paramètre de la commande **MODIFIER FORMULAIRE**. Si ce paramètre est valide, la bibliothèque peut être affichée et utilisée avec le formulaire utilisateur. Pour ouvrir la bibliothèque, l'utilisateur doit cliquer sur l'icône correspondante dans la barre d'outils de l'éditeur :



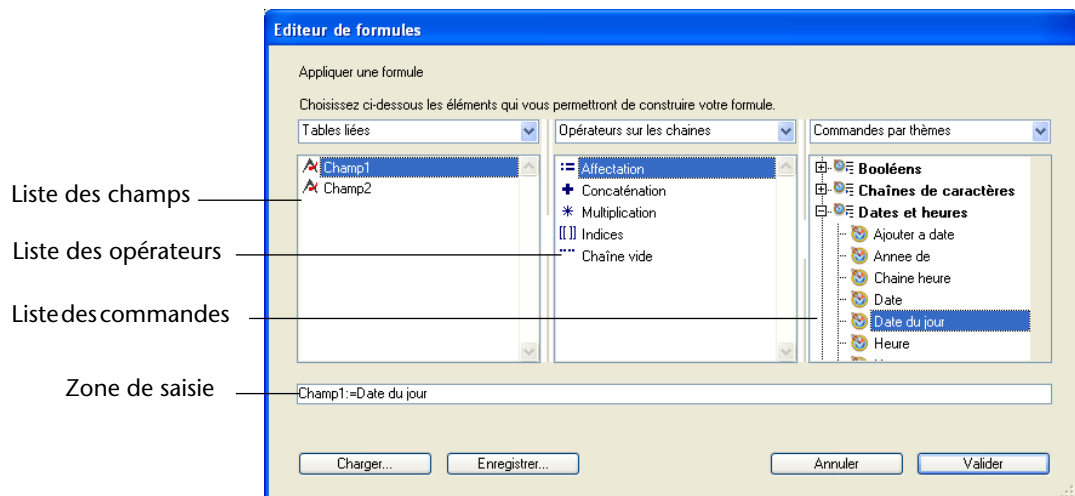
La fenêtre de la bibliothèque s'affiche alors. L'utilisateur peut ajouter des objets dans le formulaire par glisser-déposer ou copier-coller depuis la bibliothèque. Les objets sont ajoutés avec toutes leurs propriétés et actions automatiques (pour plus d'informations, reportez-vous au paragraphe "Formulaires utilisateurs et bibliothèques d'objets", page 255).

## Editeur de formules

L'éditeur de formules a été modifié dans 4<sup>e</sup> Dimension 2004. Les modifications concernent l'interface et la liste des commandes disponibles.

### Interface

L'interface de l'éditeur de formules a été modifiée de manière à rendre l'écriture des formules plus intuitive et plus logique. L'éditeur a désormais l'apparence suivante :



La saisie des formules s'effectue désormais par la sélection successive d'éléments situés dans trois listes. Le contenu de chaque liste est modifiable à l'aide du menu situé au-dessus de la liste. Il est possible de redimensionner chaque liste.

A noter également que la zone de saisie prend en charge le glisser-déposer.

- La liste des champs permet de sélectionner le champ auquel appliquer la formule. Comme dans les versions précédentes de 4<sup>e</sup> Dimension, le menu de la liste permet d'afficher les champs de la table courante, des tables liées ou de toutes les tables.

---

*Note* L'éditeur de formules tient compte désormais des noms de champs et de tables éventuellement définis par les commandes FIXER TITRES CHAMPS et FIXER TITRES TABLES (cf. page 310).

---

- La liste des opérateurs permet de choisir les opérateurs à utiliser dans la formule. Les opérateurs sont désormais classés par thèmes, auxquels vous pouvez accéder via le menu de la liste :



Chaque thème affiche tous les opérateurs disponibles pour le type de données ou d'opération correspondant. Par exemple, l'opérateur d'affectation := est disponible pour tous les types de données. Vous disposez également de nouveaux opérateurs :

- **Chaîne vide** insère deux guillemets "" (*Opérateurs sur les chaînes*)
  - **Division entière** \ (*Opérateurs sur les numériques et sur les heures*)
  - **Exponentiation** ^ (*Opérateurs sur les numériques*)
  - **Date vierge** insère !00/00/00! (*Opérateurs sur les dates*)
  - **Heure vierge** insère ?00:00:00? (*Opérateurs sur les heures*)
  - **&** (ET logique) et **|** (OU logique) (*Opérateurs logiques*)
- La liste des commandes contient les commandes utilisables dans le cadre d'une formule utilisateur ainsi que les méthodes éventuellement autorisées par le développeur (cf. paragraphe ci-dessous). Comme dans les versions précédentes de 4<sup>e</sup> Dimension, le menu de la liste permet d'afficher les commandes par thème ou par ordre alphabétique.

## Liste des commandes

Dans 4<sup>e</sup> Dimension 2004, seules les fonctions utilisables dans le contexte des formules sont affichées dans la liste des commandes et acceptées dans le texte de la formule : si l'utilisateur saisit le nom d'une commande non autorisée, la formule n'est pas exécutée. De même, seules les méthodes projet spécifiquement autorisées sont utilisables. Ce principe permet de renforcer la sécurité des données.

Si vous souhaitez donner accès à des méthodes projet, vous devez les autoriser explicitement. De même, si vous souhaitez donner accès à des commandes supplémentaires ou à des commandes de plug-ins, vous devez les insérer dans des méthodes projet qui seront ensuite autorisées dans l'éditeur. Pour cela, vous devez utiliser le mode programmé et employer les nouvelles commandes de gestion des formules. Pour plus d'informations sur ce point, reportez-vous au [paragraphe "Formules", page 251](#).

---

*Note* Ce nouveau principe peut empêcher le fonctionnement, dans les bases converties, des états rapides faisant appel à des méthodes personnalisées. Il est nécessaire dans ce cas d'autoriser les méthodes.

---

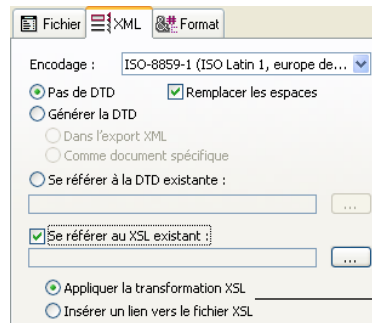
## Importation et exportation

Deux nouveautés concernent l'importation et l'exportation de données dans 4<sup>e</sup> Dimension 2004 :

- il est désormais possible d'appliquer directement une transformation XSL au moment de l'exportation de données au format XML ;
- 4<sup>e</sup> Dimension 2004 prend en charge l'importation et l'exportation de données depuis et vers une source ODBC.

## Appliquer la transformation XSL

La boîte de dialogue d'exportation de données propose désormais deux nouvelles options lorsque vous exportez des données au format XML : **Appliquer la transformation XSL** ou **Insérer un lien vers le fichier XSL**:



Nouvelles options

Ces options sont utilisables uniquement lorsqu'un fichier XSL a été désigné.

Lorsque l'option **Appliquer la transformation XSL** est cochée, 4<sup>e</sup> Dimension effectue directement la transformation XSL des données au moment de l'exportation, sur la base du document XSL désigné.

Lorsque l'option **Insérer un lien vers le fichier XSL** est cochée, 4<sup>e</sup> Dimension insère uniquement dans le document exporté la référence au XSL désigné (la transformation n'est pas effectuée).

---

*Note* De nouvelles commandes du langage permettent également de gérer les transformations XSL dans 4<sup>e</sup> Dimension 2004. Pour plus d'informations, reportez-vous au [paragraphe "Prise en charge des transformations XSL"](#), page 401.

---

## Importation et exportation via une source de données ODBC

4<sup>e</sup> Dimension 2004 vous permet d'importer et d'exporter des données depuis ou vers une source de données ODBC. Cette nouvelle fonction étend les possibilités d'échange de données entre 4<sup>e</sup> Dimension et les autres applications.

---

*Note* Dans 4<sup>e</sup> Dimension 2004, la connexion à des applications ODBC peut également être effectuée à l'aide de nouvelles commandes de langage spécifiques. Pour plus d'informations sur ce point, reportez-vous au [paragraphe "Source de données externes"](#), page 408.

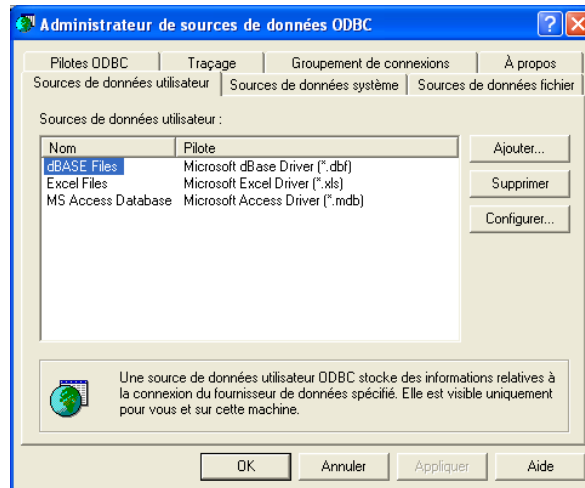
---

## Qu'est-ce qu'une source ODBC ?

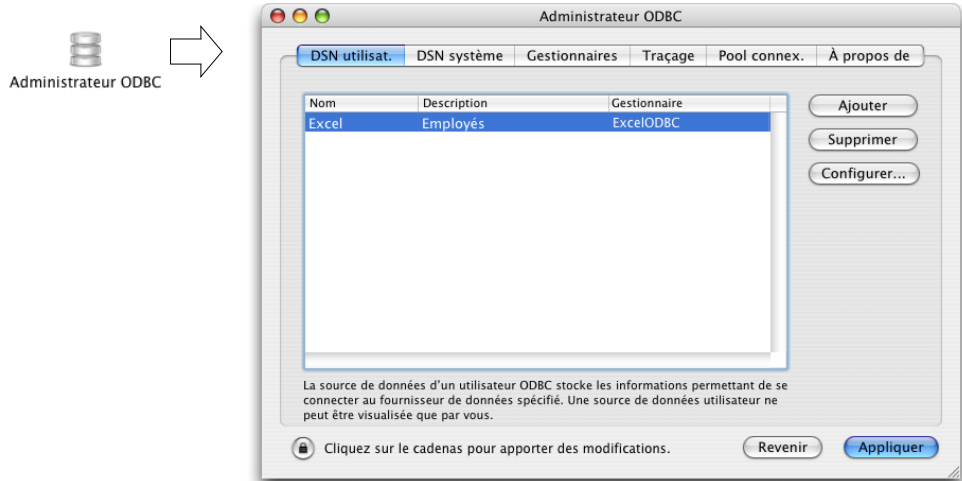
Le protocole ODBC (*Open DataBase Connectivity*) est une interface permettant à des applications différentes de communiquer entre elles via le langage SQL. Une source de données ODBC est une "fenêtre" ouverte en permanence sur un document appartenant à une application, même lorsque cette application n'est pas ouverte. De nombreuses applications (bases de données, tableurs, gestionnaires de données...) permettent la publication de données sous forme de source ODBC à l'aide d'un pilote ODBC spécifique.

Toute application prenant en charge le protocole ODBC peut lire ou écrire des données dans une source ODBC. La fonction d'importation de 4<sup>e</sup> Dimension permet de récupérer des données depuis une source ODBC. La fonction d'exportation de 4<sup>e</sup> Dimension permet d'insérer des données dans la source ODBC.

- Sous Windows, la définition d'une source ODBC s'effectue via le panneau de configuration **Outils d'administration puis Sources de données ODBC** :



- Sous MacOS X, la définition d'une source ODBC s'effectue à l'aide du programme **Administrateur ODBC**, situé dans les **Utilitaires** :

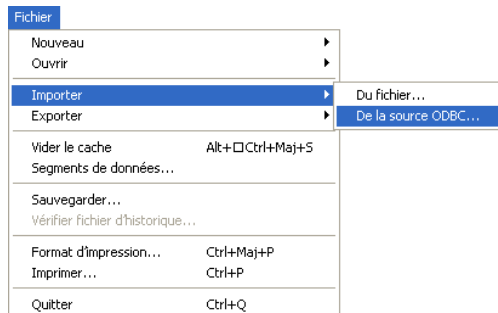


*Note* Pour plus d'informations sur ces boîtes de dialogue, reportez-vous à la documentation de votre système d'exploitation.

### Importer des données depuis une source ODBC

Le principe de l'importation de données depuis une source ODBC est identique à celui des autres types de données, à l'exception du fait qu'une boîte de dialogue intermédiaire de sélection de source ODBC est utilisée.

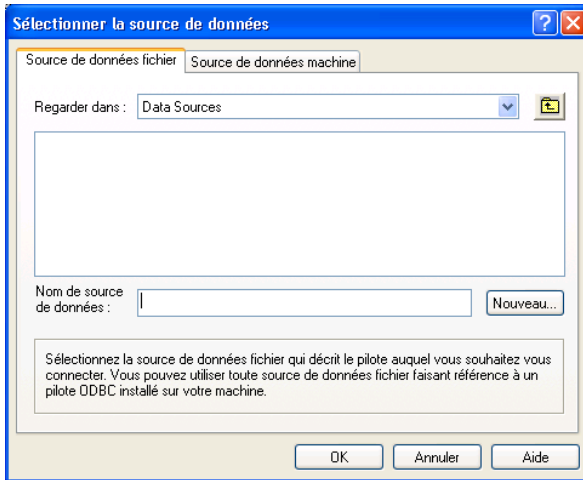
- Pour importer des données depuis une source ODBC :
- 1 Choisissez la commande **Importer>De la source ODBC...** dans le menu **Fichier** de 4<sup>e</sup> Dimension :



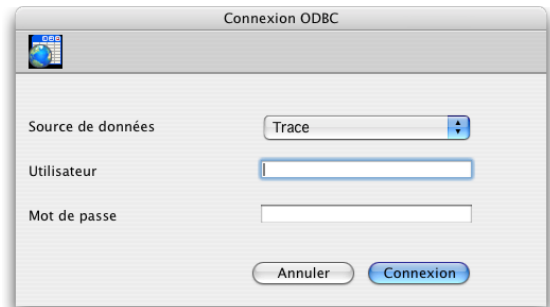


La boîte de dialogue de sélection de la source de données ODBC apparaît :

Windows



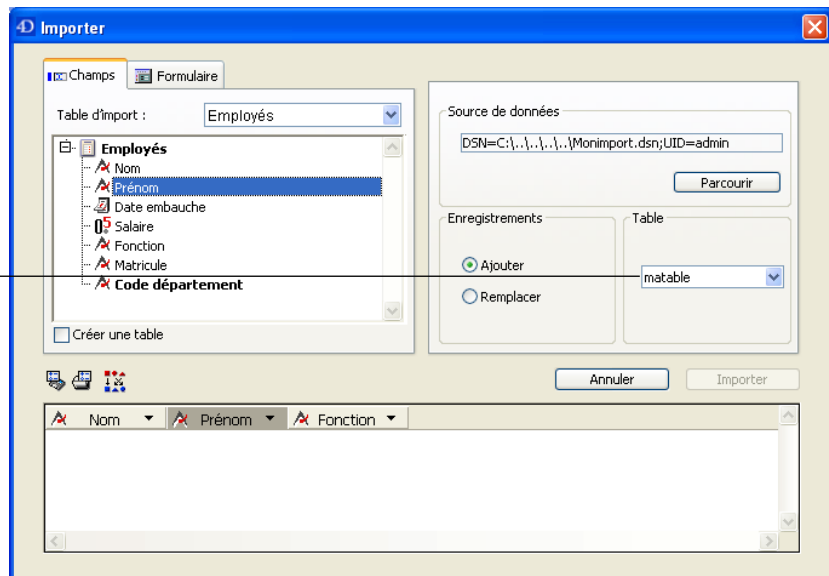
MacOS



**2 Sélectionnez la source de données à utiliser puis cliquez sur OK (Windows) ou Connexion (MacOS).**

La boîte de dialogue standard d'importation de données de 4<sup>e</sup> Dimension apparaît alors :

Sélection de table SQL



- 3 Dans la partie droite de la boîte de dialogue, désignez (à l'aide du menu "Sélectionnez une table") la table SQL de laquelle vous souhaitez importer les données.
- 4 Sélectionnez la table et les champs 4D dans lesquels les données doivent être importées ou cochez l'option "Créer une table" (option non disponible dans les bases compilées).

Comme pour les autres types d'importations, la zone de prévisualisation affiche les correspondances entre les données importées et les champs. Vous pouvez modifier ces correspondances.

---

*Note* Vous pouvez également cliquer sur l'onglet **Formulaire** et importer les données via un formulaire.

---

- 5 Choisissez les options d'importation, dans la partie droite de la boîte de dialogue :
  - **Source de données** : cette zone décrit la source de données à laquelle la base est connectée. Vous pouvez afficher de nouveau la boîte de dialogue de sélection de source de données en cliquant sur le bouton **Parcourir**.
  - **Enregistrements** : vous pouvez ajouter ou remplacer les enregistrements de la table d'importation avec les données provenant de la source.
- 6 Cliquez sur le bouton **Import** pour importer les données.

Une fois les données importées, la connexion entre 4<sup>e</sup> Dimension et la source de données est automatiquement refermée.

Si vous le souhaitez, vous pouvez sauvegarder tous les paramètres de l'importation dans un fichier de projet afin de pouvoir le charger par la suite via la boîte de dialogue ou la commande **ODBC IMPORTER**.

Pour cela, cliquez sur le bouton d'enregistrement des préférences **Enregistrer préférences**, situé en bas à gauche de la boîte de dialogue.

---

*Note* Les projets sauvegardés dans la boîte de dialogue d'importation ODBC ne sont pas compatibles avec la boîte de dialogue d'importation standard de 4<sup>e</sup> Dimension.

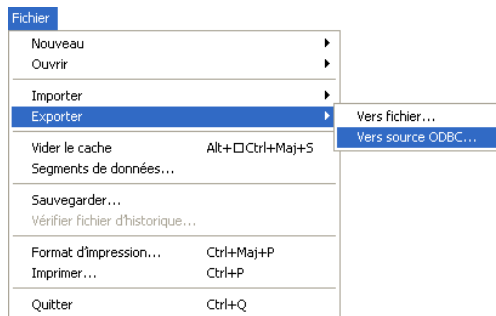
---

### Exporter des données vers une source ODBC

Le principe de l'exportation de données 4D vers une source ODBC est semblable à celui des autres types de données, à l'exception du fait qu'une boîte de dialogue intermédiaire de sélection de source ODBC est utilisée. En outre, les données exportées ne sont pas stockées dans un fichier disque mais directement dans la source ODBC publiée.

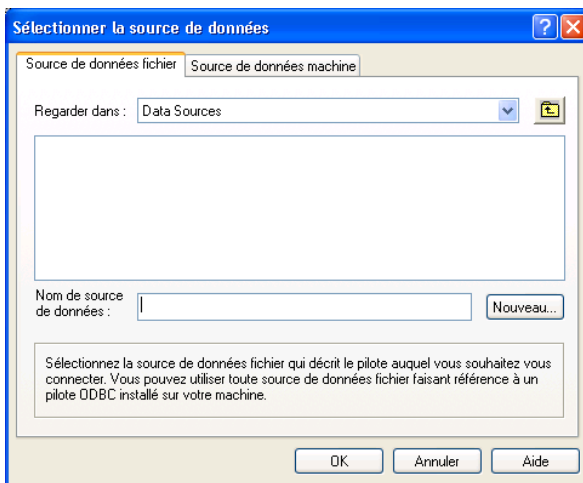
- Pour exporter des données vers une source ODBC :

**1 Choisissez la commande Exporter>Vers la source ODBC... dans le menu Fichier de 4<sup>e</sup> Dimension :**

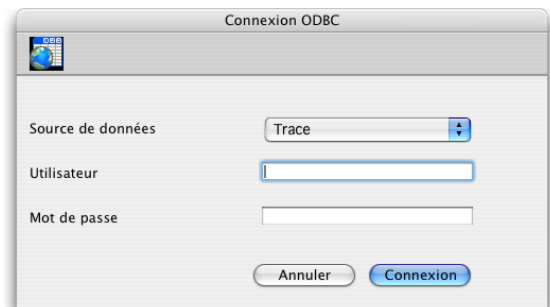


La boîte de dialogue de sélection de la source de données ODBC apparaît :

Windows



MacOS



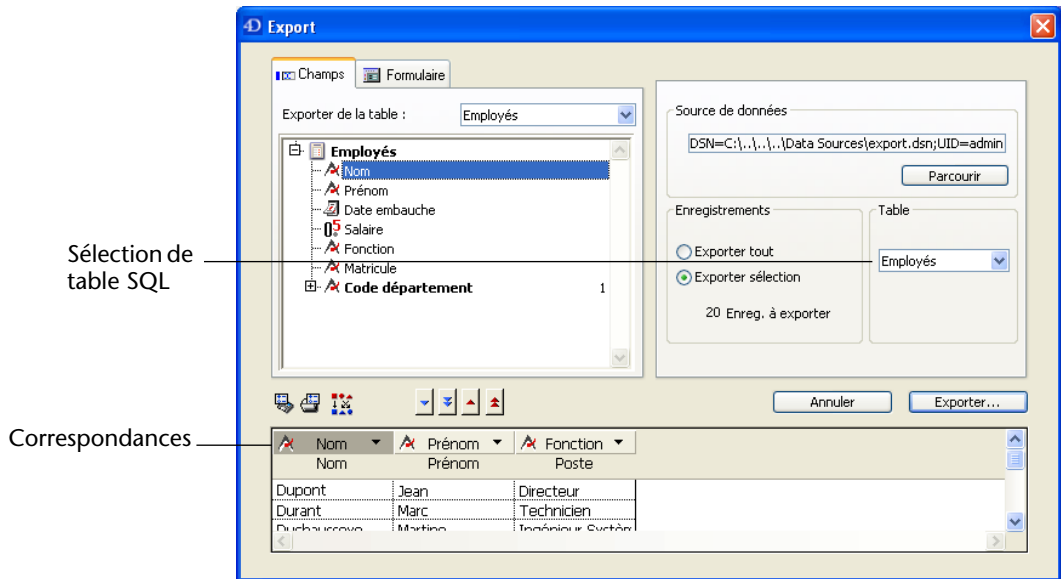
**2 Sélectionnez la source de données à utiliser puis cliquez sur OK (Windows) ou Connexion (MacOS).**

La boîte de dialogue standard d'exportation de données de 4<sup>e</sup> Dimension apparaît.

**3 Dans la partie droite de la boîte de dialogue, désignez (à l'aide du menu "Sélectionnez une table") la table SQL dans laquelle vous souhaitez exporter les données.**

**4 Sélectionnez la table et les champs 4D desquels les données doivent être exportées.**

Comme pour les autres types d'exportations, la zone de prévisualisation affiche les correspondances entre les données exportées et les champs. Vous pouvez modifier ces correspondances.



*Note* Vous pouvez également cliquer sur l'onglet **Formulaire** et exporter les données à l'aide d'un formulaire.

**5 Choisissez les options d'exportation dans la partie droite de la boîte de dialogue :**

- **Source de données** : cette zone décrit la source de données à laquelle la base est connectée. Vous pouvez afficher de nouveau la boîte de dialogue de sélection de source de données en cliquant sur le bouton **Parcourir**.
- **Enregistrements** : vous pouvez exporter uniquement la sélection courante ou tous les enregistrements de la table.

**6 Cliquez sur le bouton Exporter... pour exporter les données vers la source ODBC.**

Une fois les données exportées, la connexion entre 4<sup>e</sup> Dimension et la source de données est automatiquement refermée.

Si vous le souhaitez, vous pouvez sauvegarder tous les paramètres de l'exportation dans un fichier de projet afin de pouvoir le charger par la suite via la boîte de dialogue ou la commande **ODBC EXPORTER**. Pour cela, cliquez sur le bouton **Enregistrer préférences**, situé en bas à gauche de la boîte de dialogue.

*Note* Les projets sauvegardés dans la boîte de dialogue d'exportation ODBC ne sont pas compatibles avec la boîte de dialogue d'exportation standard de 4<sup>e</sup> Dimension.

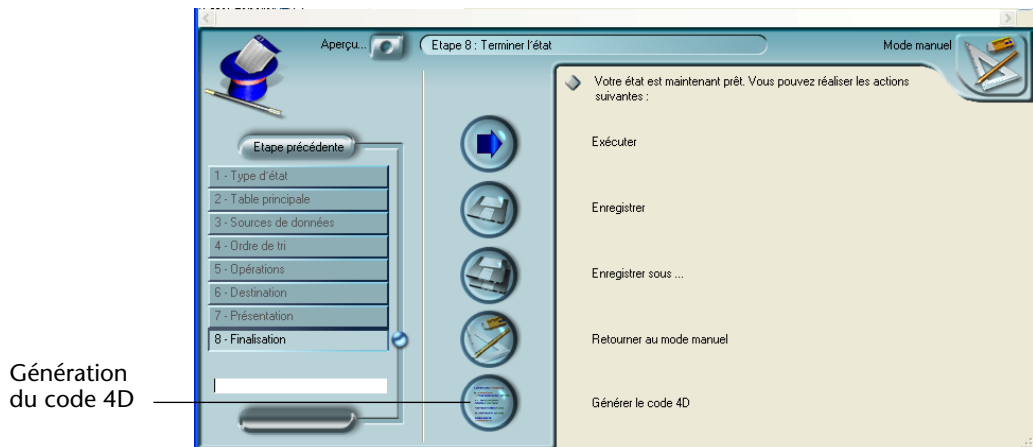
## Editeur d'états rapides

Deux principales nouveautés ont été ajoutées dans l'éditeur d'états rapides de 4<sup>e</sup> Dimension 2004 : la possibilité de générer automatiquement le code de l'état et le nouvel opérateur Ecart type.

**Génération du code** Un nouveau bouton dans l'assistant de l'éditeur d'états rapides permet de générer automatiquement le code 4<sup>e</sup> Dimension correspondant à l'état défini et de le placer dans le Presse-papiers. Cette fonction permet de construire manuellement des ébauches d'états rapides qu'il sera possible de personnaliser à l'aide du langage de 4<sup>e</sup> Dimension.

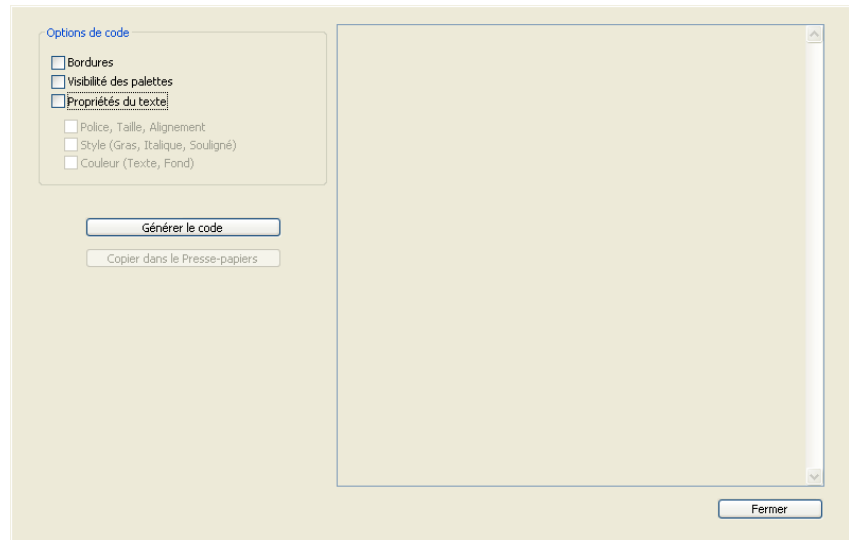
Cette fonction n'est disponible qu'en mode Utilisation.

Le bouton permettant de générer le code 4D se situe dans la dernière page de l'assistant de création d'états rapides :



*Note* Même si vous avez construit manuellement un état rapide, vous pouvez utiliser cette fonction en affichant l'étape "Finalisation" de l'assistant.

Lorsque vous cliquez sur ce bouton, une boîte de dialogue apparaît, vous permettant de paramétrer la génération du code :

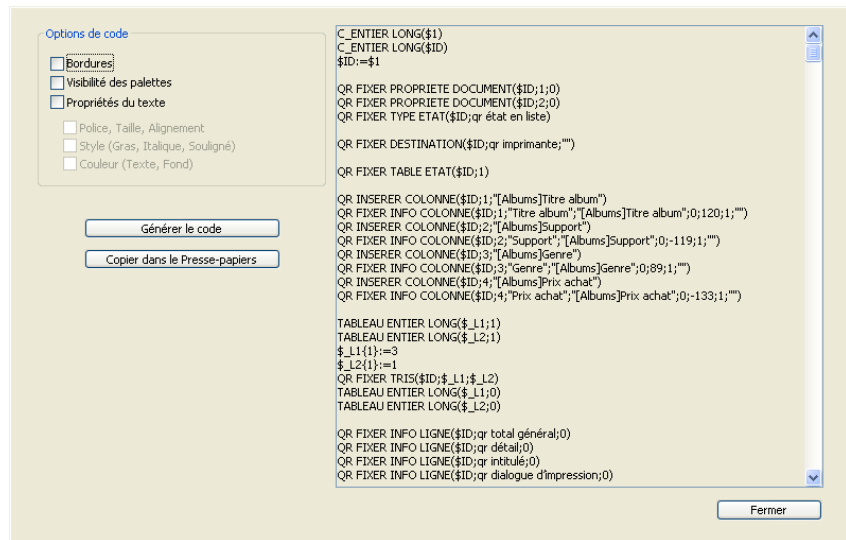


Les options suivantes sont disponibles :

- **Bordures** : cette option permet d'inclure dans le code 4D généré les propriétés d'encadrement définies pour les cellules de l'état.
- **Visibilité des palettes** : cette option permet d'inclure dans le code 4D généré la configuration d'affichage courante des différentes barres d'outils de l'éditeur (Standard, Style, Colonnes, etc.).  
Il est recommandé de désélectionner cette option si le code est destiné à une exploitation dans une zone hors écran.
- **Propriétés du texte** : cette option permet de définir si vous souhaitez intégrer ou non le paramétrage du texte de l'état (police, style, etc.) dans le code généré.  
Les trois options suivantes n'ont pas d'effet si cette option n'est pas préalablement cochée.
  - **Police, taille, alignement** : lorsque cette option est cochée, le code 4D généré inclut les informations relatives à la police, la taille de police et l'alignement des textes dans le modèle d'état.

- **Style (gras, italique, souligné)** : lorsque cette option est cochée, le code 4D généré inclut les informations relatives aux styles des textes dans le modèle d'état.
- **Couleur (texte, fond)** : lorsque cette option est cochée, le code 4D généré inclut les informations relatives à la couleur des textes dans le modèle d'état.

Une fois les options définies, vous pouvez cliquer sur le bouton **Générer le code** afin de créer le code dans la zone de prévisualisation de la fenêtre :



Vous pouvez modifier une ou plusieurs options de code et cliquer de nouveau sur **Générer le code** afin de mettre à jour la zone de prévisualisation.

Si le contenu généré vous convient, cliquez sur le bouton **Copier dans le Presse-papiers** afin de placer le code dans le Presse-papiers. Cliquez ensuite sur le bouton **Fermer** afin de refermer la fenêtre.

Vous pouvez ensuite coller le code dans une méthode ou fichier texte à l'aide du raccourci standard **Ctrl+v** (Windows) ou **Commande+v** (MacOS) ou de la commande **Coller** du menu **Edition**.

## Ecart type

Un nouvel opérateur est disponible dans le menu **Cellules** et dans la palette de l'éditeur d'états rapides : **Ecart type**.



Ecart type

L'écart type d'une série de valeurs est la racine carrée de la variance (la variance est une valeur de dispersion autour de la moyenne). Pour plus d'informations sur l'écart type et la variance, reportez-vous au thème "Fonctions statistiques" du manuel *Langage* de 4<sup>e</sup> Dimension.

L'accès par programmation à cet opérateur s'effectue via deux nouvelles constantes :

- qr cmd écart type (valeur 513) dans thème "QR Commandes" pour les commandes QR EXECUTER COMMANDE et QR Lire statut commande,
- qr écart type (valeur 32) dans le thème "QR Opérateurs" pour les commandes QR FIXER DONNEES TOTAUX et QR LIRE DONNEES TOTAUX.

Le code ##D doit être utilisé avec la commande QR FIXER DONNEES TOTAUX si vous souhaitez passer à la fois du texte et des opérateurs dans le paramètre *valeur*.



# 4

## Mode Structure

De nombreuses modifications, aussi bien graphiques que fonctionnelles, ont été apportées à l'environnement de développement de 4<sup>e</sup> Dimension 2004 :

- Possibilité pour l'utilisateur de modifier son mot de passe et d'enregistrer son accès depuis la boîte de dialogue d'identification.
- Nouveaux paramétrages dans la boîte de dialogue des Préférences.
- Regroupement des différents éditeurs dans une nouvelle "Boîte à outils" : utilisateurs et groupes, menus, bibliothèque d'images, messages d'aide, énumérations, feuilles de style, formats et filtres.
- Modernisation générale de l'interface du mode Structure : nouvelle barre d'outils, nouvelle fenêtre Inspecteur dans l'éditeur de Structure, nouvelle apparence de l'Explorateur...
- Gestion de dossiers d'objets dans l'Explorateur.
- Nouvelles possibilités dans l'éditeur de méthodes.
- Modifications de l'éditeur de formulaires : nouvelle interface, nouveaux outils (vues, zoom, aides à l'alignement, etc.), nouvelle liste des propriétés...
- Modifications des objets de formulaires : nouvel objet "List box", séparateurs, interface, contrôles de saisie natifs, case à cocher à trois états, boutons 3D...
- Harmonisation de la gestion des listes d'enregistrements.
- Possibilité de définir des formulaires modifiables par l'utilisateur et de mettre à sa disposition une bibliothèque d'objets.
- Nouveau générateur d'applications.
- Web Services : meilleure prise en charge des services DOC (souscription et publication).

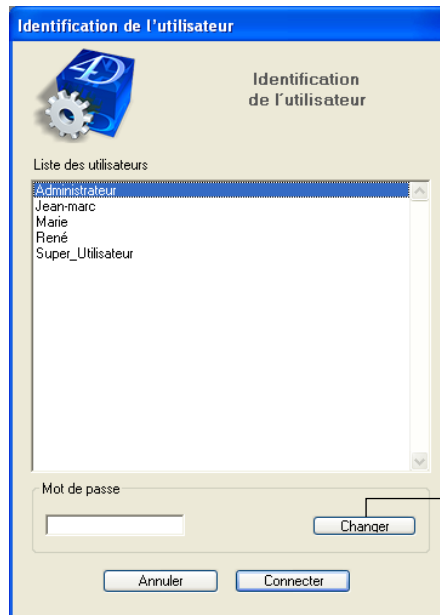
## Boîte de dialogue d'identification

La boîte de dialogue standard d'identification de l'utilisateur a été modifiée :

- un nouveau bouton permet à l'utilisateur de modifier à la volée son mot de passe.
- un nouveau bouton permet à l'utilisateur de sauvegarder les paramètres d'accès à la base dans un fichier (4D Client uniquement).

### Modification du mot de passe

La boîte de dialogue d'identification de l'utilisateur (qui apparaît lors de l'ouverture d'une base protégée par mot de passe) comporte désormais le bouton **Changer** :



Modification du mot de passe

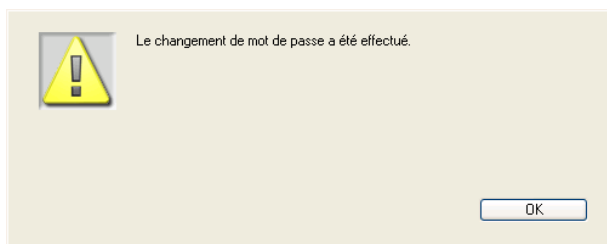
Ce bouton permet d'accéder à la boîte de dialogue de modification du mot de passe de l'utilisateur courant. Si la liste des utilisateurs est affichée, l'utilisateur courant est l'utilisateur sélectionné. Si la liste n'est pas affichée, le nom de l'utilisateur courant doit être saisi dans la boîte de dialogue d'identification (l'affichage de la liste des utilisateurs est paramétré dans les Préférences de la base).

Lorsque l'utilisateur clique sur le bouton **Changer**, la boîte de dialogue suivante apparaît :

Pour pouvoir modifier son mot de passe, l'utilisateur doit au préalable indiquer son ancien mot de passe. Le nouveau mot de passe doit être saisi puis confirmé.

*Note* Il est possible de mettre en place un système de contrôle des mots de passe utilisateurs dans 4<sup>e</sup> Dimension 2004 à l'aide de boîtes de dialogue personnalisées. Pour plus d'informations sur ce point, reportez-vous au [paragraphe "Proposer une boîte de dialogue de gestion d'accès personnalisée", page 279.](#)

Une fois la boîte de dialogue validée, si la saisie est correcte, 4<sup>e</sup> Dimension indique que la modification a eu lieu :



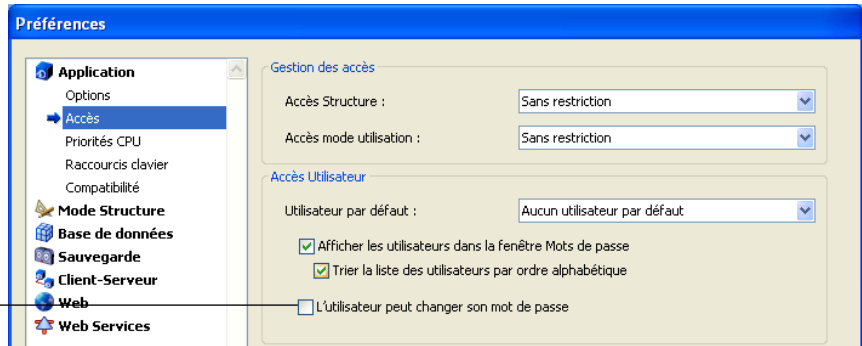
Le nouveau mot de passe de l'utilisateur remplace l'ancien dans la base de données en mode Structure, l'utilisateur n'a plus qu'à cliquer sur le bouton **Connecter** pour ouvrir la base.

**Ne pas afficher le bouton Changer**

Si vous le souhaitez, vous pouvez masquer le bouton **Changer** afin de ne pas permettre aux utilisateurs de modifier leur mot du passe. Il suffit

pour cela de désélectionner l'option "L'utilisateur peut changer son mot de passe" dans la page **Application/Accès** des Préférences de la base :

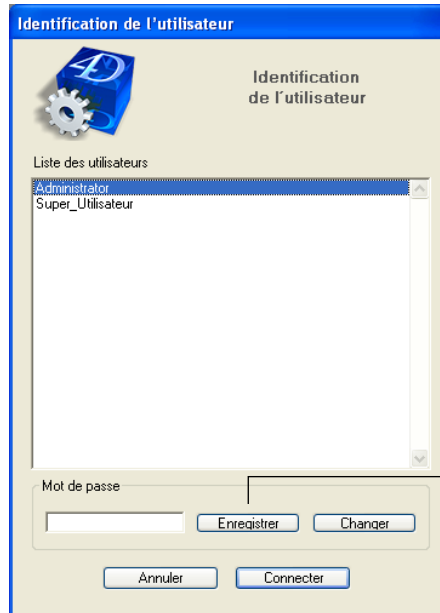
Affichage du bouton  
Changer



Pour des raisons de compatibilité, le bouton est masqué par défaut dans les bases de données créées avec une version précédente de 4<sup>e</sup> Dimension et converties en version 2004.

### Sauvegarde des paramètres d'accès à la base

La boîte de dialogue d'identification de l'utilisateur de 4D Client 2004 comporte le nouveau bouton **Enregistrer** :



Enregistrement des  
paramètres d'accès

---

*Note* Cette boîte de dialogue apparaît juste après la sélection de la base 4D Server, lorsque le système de protection par mots de passe est activé.

---

Ce bouton permet d'enregistrer les paramètres d'accès à la base 4D Server dans un fichier de connexion 4D Client.

Lorsque vous glissez-déposez un fichier de connexion sur l'application 4D Client (ou double-cliquez sur le fichier), vous accédez directement à la base 4D Server, sans qu'il soit nécessaire de la désigner dans la boîte de dialogue de connexion. En outre, le fichier de connexion peut contenir le mot de passe utilisateur (crypté). Dans ce cas, l'accès à la base 4D Server est immédiat.

---

*Note* Dans les versions précédentes de 4D Client, cette fonction était accessible via l'éditeur de mots de passe sur le poste 4D Client.

---

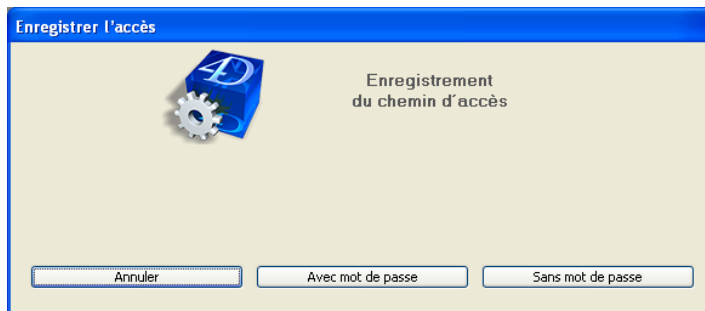
► Pour enregistrer les paramètres d'accès à la base :

**1 Dans la boîte de dialogue d'identification de l'utilisateur, sélectionnez ou saisissez (suivant le paramétrage courant) le nom d'utilisateur pour lequel enregistrer l'accès, puis saisissez son mot de passe.**

Ces paramètres doivent être valides pour que vous puissiez accéder à la boîte de dialogue d'enregistrement du fichier de connexion.

**2 Cliquez sur le bouton Enregistrer.**

La boîte de dialogue suivante apparaît :



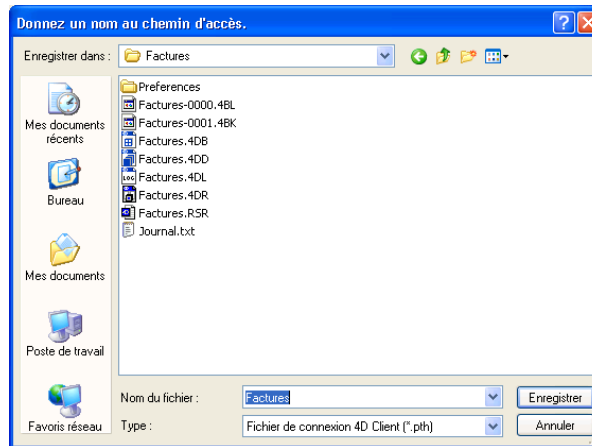
Vous pouvez enregistrer le chemin d'accès avec ou sans mot de passe.

- **Avec mot de passe** : dans ce cas, le chemin d'accès de la base et le mot de passe (crypté) de l'utilisateur sont stockés dans le document. Au lancement de 4D Client, aucune boîte de dialogue n'apparaît, la connexion est immédiate.

- **Sans mot de passe** : dans ce cas, le chemin d'accès de la base est stocké mais l'utilisateur devra saisir son mot de passe pour pouvoir l'ouvrir.

### 3 Cliquez sur le bouton Avec mot de passe ou Sans mot de passe.

Une boîte de dialogue standard d'enregistrement de fichier apparaît, vous permettant de désigner le nom et l'emplacement du document sur votre disque. L'extension des fichiers de connexion est ".pth".



### 4 Choisissez le nom et l'emplacement du fichier et validez la boîte de dialogue d'enregistrement.

Le fichier de connexion 4D Client est stocké sur disque.


## Préférences




La boîte de dialogue des Préférences a été réorganisée et enrichie afin de contenir les nouvelles fonctions de 4<sup>e</sup> Dimension 2004 (sauvegarde, compatibilité, etc.) ainsi que les options issues de l'intégration de l'utilitaire 4D Customizer Plus.

En effet, 4D Customizer Plus n'est plus disponible avec la gamme 4<sup>e</sup> Dimension 2004. Les options auparavant présentes dans cet utilitaire sont soit intégrées dans les Préférences de 4<sup>e</sup> Dimension, soit supprimées car inutiles ou redondantes.




## Tableau de comparaison

Le tableau suivant détaille les paramètres des Préférences de 4<sup>e</sup> Dimension 2004 et précise leur provenance (page des Préférences 4<sup>e</sup> Dimension 2003, 4D Customizer Plus ou nouveauté) :

Préférences 4 <sup>e</sup> Dimension 2004			Statut précédent
Thème	Page	Préférences	
Application 	Options	Mode de démarrage	4D2003 Mode Structure/Options
		Emplacement du dossier temporaire	4D2003 Application/Ressources système
		Zone de déposer	4D2003 Interface/Apparence
		Progression	4D2003 Interface/Apparence
		Afficher la barre d'outils <sup>1</sup>	4D2003 Interface/Apparence
		Afficher fenêtres	4D Customizer Plus, voir <a href="#">paragraphe "Afficher fenêtres"</a> , page 65
	Accès	Gestion des accès	4D2003 Base de données/Gestion des accès
		Accès Utilisateur	4D2003 Base de données/Gestion des accès + <i>Nouveauté 4D 2004</i> , voir <a href="#">paragraphe "Ne pas afficher le bouton Changer"</a> , page 51
	Priorités CPU	Définir priorité CPU	4D2003 Application/Ressources système
	Raccourcis clavier	Touches	4D Customizer Plus, voir <a href="#">paragraphe "Raccourcis clavier"</a> , page 66
	Compatibilité	Compatibilité Structure	4D2003 Mode Structure/Options + 4D2003 Interface/Apparence + <i>Nouveauté 4D 2004</i> , voir <a href="#">paragraphe "Nouvelles options de compatibilité"</a> , page 58
		Compatibilité Web	4D2003 Web/Configuration + <i>Nouveauté 4D 2004</i> , voir <a href="#">paragraphe "Nouvelles options de compatibilité"</a> , page 58
		Plate-forme <sup>2</sup>	4D2003 Interface/Apparence

Préférences 4 <sup>e</sup> Dimension 2004			Statut précédent
Thème	Page	Préférences	
Mode Structure 	Structure	Police générale	4D2003 Mode Structure/Polices
		Commentaires par défaut	4D2003 Mode Structure/Commentaires
		Editeur de structure	4D2003 Mode Structure/Editeur de structure
	Editeur de formulaires	Modèles d'objets	<i>Nouveautés 4D 2004</i> , voir <a href="#">paragraphe "Editeur de formulaires"</a> , page 60
		Déplacement	
		Magnétisme	
		Affichage par défaut	
	Editeur de méthodes	Police	4D2003 Mode Structure/Editeur de méthodes
		Affichage par défaut	4D2003 Mode Structure/Editeur de méthodes
		Options	4D2003 Mode Structure/Editeur de méthodes
		Styles des éléments de syntaxe	4D2003 Mode Structure/Editeur de méthodes
	Compilateur	Options de compilation	4D2003 Compilation/Paramétrage
Méthodes Compilateur pour		4D2003 Compilation/Paramétrage	
Documentation	Accès documentation...	4D2003 Mode Structure/Documentation	
Base de données 	Gestion des données	Paramètres généraux	4D2003 Base de données/Gestion des données + 4D2003 Application/Ressources système
		Paramètres du cache pour la base	4D2003 Application/Ressources système + <i>Nouveauté 4D 2004</i> , voir <a href="#">paragraphe "Gestion de la mémoire cache"</a> , page 61
		WEDD	<i>4D Customizer Plus</i> , voir <a href="#">paragraphe "WEDD"</a> , page 66
	Script Manager	Script Manager	<i>4D Customizer Plus</i> , voir <a href="#">paragraphe "Script Manager"</a> , page 67
Sauvegarde 	Configuration		<i>Nouveauté 4D 2004</i> , voir <a href="#">chapitre "Module de sauvegarde intégré"</a> , page 197
	Périodicité		
	Sauvegarde		
	Restitution		



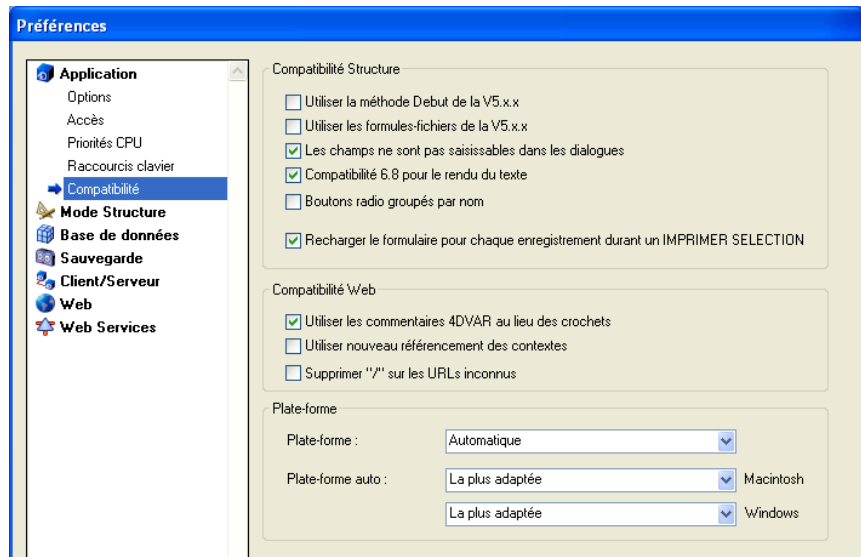
Préférences 4 <sup>e</sup> Dimension 2004			Statut précédent
Thème	Page	Préférences	
Client/ Serveur 	Configuration	Réseau	4D Customizer Plus, voir <a href="#">paragraphe "Raccourcis clavier"</a> , page 66
		Délai avant déconnexion...	4D2003 Application/Client/Serveur
		Communication Client-Serveur	4D2003 Application/Client/Serveur + <i>Nouveauté 4D 2004</i> , voir <a href="#">paragraphe "Reconnexion automatique des clients"</a> , page 64
		4D Open	4D2003 Application/Client/Serveur
	Publication	Information de publication	4D Customizer Plus, voir <a href="#">paragraphe "Publication"</a> , page 69
		Table de configuration Accepter-refuser	4D Customizer Plus, voir <a href="#">paragraphe "Publication"</a> , page 69
Cryptage		4D2003 Application/Client/Serveur	
Web 	Configuration	Publication du serveur Web	4D2003 Web/Publication + <i>Nouveauté 4D 2004</i> , voir <a href="#">paragraphe "Modifier le numéro de port HTTPS"</a> , page 243
		Chemin d'accès HTML...	4D2003 Web/Publication
		Mode de démarrage	4D2003 Web/Publication
		Avancé	Mémoire cache
	Process Web		4D2003 Web/Configuration
	Options		4D2003 Web/Configuration
	Mots de passe Web		4D2003 Web/Publication
	Options	Conversion texte	4D2003 Web/Configuration
		4D WebSTAR	4D2003 Web/4D WebSTAR
		Connexions persistantes	<i>Nouveauté 4D 2004</i> , voir <a href="#">paragraphe "Connexions persistantes"</a> , page 241
Web Services 	SOAP	Serveur	4D2003 Web/Web Services
		Client	4D2003 Web/Web Services

1. Il est désormais possible d'afficher ou de masquer la barre d'outils en mode Menus créés uniquement.
2. La gestion de l'interface de plate-forme a été modifiée dans 4<sup>e</sup> Dimension 2004, reportez-vous au [paragraphe "Interface de plate-forme"](#), page 148.

Les principales nouveautés sont détaillées dans les pages suivantes.

## Nouvelles options de compatibilité

La page **Compatibilité** du thème “Applications” propose de nouvelles options de compatibilité :



### ■ Les champs ne sont pas saisissables dans les dialogues

dans les versions précédentes de 4<sup>e</sup> Dimension, il n’était pas possible de saisir des valeurs par l’intermédiaire de champs dans des boîtes de dialogue (affichées par exemple à l’aide de la commande DIALOGUE). Cette limitation a été supprimée dans 4<sup>e</sup> Dimension 2004.

Vous pouvez toutefois souhaiter conserver le fonctionnement précédent, notamment si votre base de données utilise des champs dans des boîtes de dialogue pour afficher des données.

Par défaut, cette option est cochée pour les anciennes bases de données converties en version 2004 et est désélectionnée pour les bases de données créées en version 2004.

### ■ Boutons radio groupés par nom

Cette option conserve l’ancien mode de regroupement fonctionnel des boutons radio, basé sur leur lettre initiale. Pour plus d’informations, reportez-vous au [paragraphe “Nouveau mode de fonctionnement des boutons radio”](#), page 164.

Cette option n’apparaît que pour les anciennes bases de données converties en version 2004. Elle est cochée par défaut.

Les bases de données créées en version 2004 utilisent le nouveau mode.

### ■ Recharger le formulaire pour chaque enregistrement durant un IMPRIMER SELECTION

Dans les versions précédentes de 4<sup>e</sup> Dimension, le formulaire utilisé lors d'une impression lancée à l'aide de la commande IMPRIMER SELECTION était rechargé pour chaque enregistrement. 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>e</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**.

Cette option n'apparaît que pour les anciennes bases de données converties en version 2004. Elle est cochée par défaut.

Les bases de données créées en version 2004 utilisent le nouveau mode.

### ■ Supprimer "/" sur les URLs inconnus

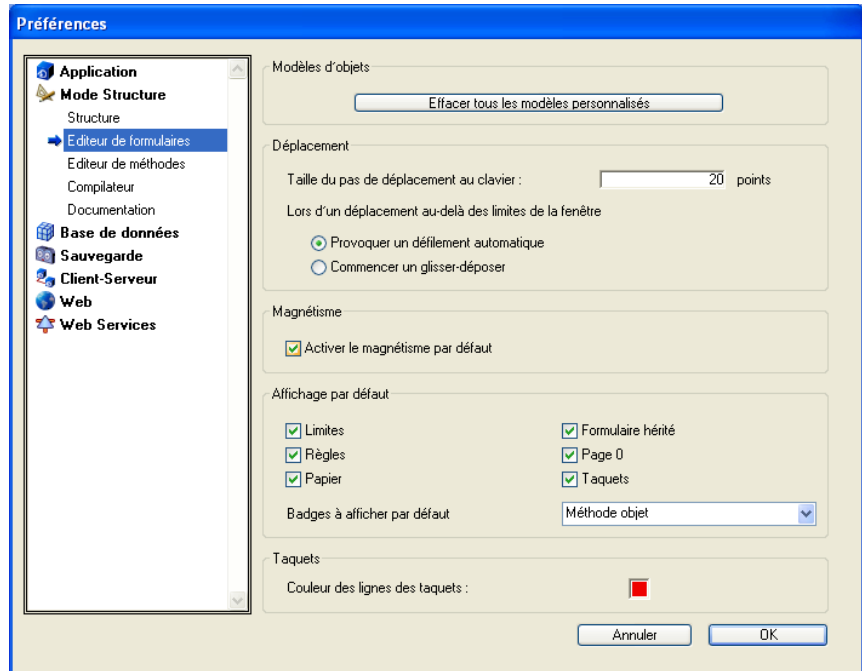
Dans les versions précédentes de 4<sup>e</sup> Dimension, les URLs inconnus (URLs ne correspondant à aucune page ni URLs spécial) étaient retournés dans les méthodes bases Sur authentification Web et Sur connexion Web (\$1) sans débiter par le caractère "/". Cette particularité a été supprimée dans 4<sup>e</sup> Dimension 2004 (cf. [paragraphe "Prise en charge des URLs inconnus", page 244](#)).

Toutefois, si vous avez mis en place des mécanismes basés sur cette particularité et souhaitez conserver le fonctionnement précédent, vous pouvez cocher l'option **Supprimer "/" sur les URLs inconnus**.

Par défaut, cette option est cochée pour les bases de données converties et non cochée pour les nouvelles bases.

## Editeur de formulaires

La page **Editeur de formulaires** (thème “Mode Structure”) propose plusieurs nouvelles options permettant de définir le fonctionnement par défaut de l’éditeur.



- **Effacer tous les modèles personnalisés**  
Ce bouton permet de réinitialiser tous les modèles d’objet à leur valeur par défaut. Pour plus d’informations, reportez-vous au [paragraphe “Utiliser comme modèle”, page 128](#).
- **Taille du pas de déplacement**  
Cette zone permet de définir la valeur (en pixels) du pas utilisé pour le déplacement ou le redimensionnement d’un objet par le clavier et avec la touche **Majuscule**.
- **Lors d’un déplacement au-delà des limites de la fenêtre**  
Cette option permet de définir le mode de fonctionnement de l’éditeur de formulaires lors du déplacement d’un objet à l’aide de la souris au-delà des limites de la fenêtre du formulaire.
  - **Provoquer un défilement automatique** : lorsque cette option est cochée, cette action provoque le défilement du formulaire dans la fenêtre, comme si vous aviez cliqué sur les barres de défilement. Ce fonctionnement est utile pour déplacer des objets dans des formulaires de grande taille.

- **Commencer un glisser-déposer** : lorsque cette option est cochée, cette action est interprétée comme un glisser-déposer. La fenêtre du formulaire n'est pas modifiée et l'objet déplacé peut être déposé dans une autre fenêtre (si son contenu est compatible), par exemple un autre formulaire. Ce fonctionnement est utile pour recycler des objets entre plusieurs formulaires.

Vous pouvez configurer cette option en fonction de vos habitudes de travail et de vos besoins en développement. Par défaut, l'option de défilement automatique (fonctionnement traditionnel de 4<sup>e</sup> Dimension) est sélectionnée.

- **Activer le magnétisme par défaut**  
Cette option permet d'activer par défaut le magnétisme dans chaque nouvelle fenêtre de l'éditeur de formulaires. Il est possible de modifier cette option individuellement dans chaque fenêtre (cf. [paragraphe "Utiliser la grille magnétique", page 129](#)).
- **Affichage par défaut**  
Ces options permettent de définir les éléments affichés ou masqués par défaut dans chaque nouvelle fenêtre de l'éditeur de formulaires. Il est possible de modifier séparément l'affichage de chaque fenêtre à l'aide du menu hiérarchique **Afficher** de l'éditeur de formulaires.  
Vous pouvez également définir les badges à afficher par défaut. Pour plus d'informations sur les badges, reportez-vous au [paragraphe "Utilisation des badges", page 125](#).
- **Couleur des lignes des taquets**  
Cette nouvelle option permet de modifier la couleur des lignes des taquets utilisés dans l'éditeur de formulaires pour définir les différentes zones (en-tête, ruptures, corps et pied de page, etc.).

## Gestion de la mémoire cache

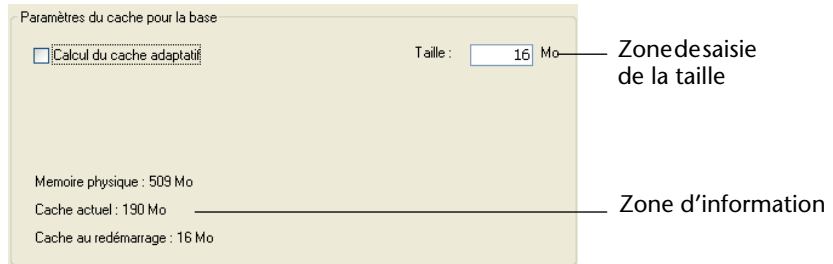
**A propos de la mémoire principale** : la gestion de la mémoire principale a été optimisée dans 4<sup>e</sup> Dimension 2004, conformément aux spécifications de Windows XP et MacOS X. Désormais, la mémoire utilisée par 4<sup>e</sup> Dimension est automatiquement allouée par le système, il n'est plus possible de la personnaliser. En revanche, il est toujours possible de personnaliser la mémoire cache de vos bases de données.

Le paramétrage de la mémoire cache a été modifié dans 4<sup>e</sup> Dimension 2004. Désormais, vous disposez de deux modes de calcul du cache :

- le mode "forcé"
- le mode adaptatif (recommandé et utilisé par défaut)

### Mode "forcé"

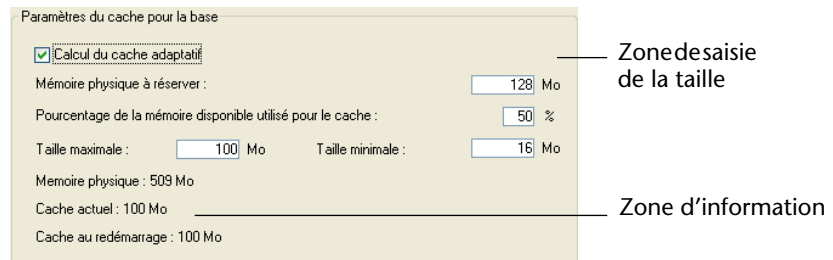
Dans ce mode, vous définissez vous-même la taille de la mémoire cache pour la base. Pour cela, il vous suffit de désélectionner l'option **Calcul du cache adaptatif** dans la page **Gestion des données** (thème "Base de données"). 4<sup>e</sup> Dimension affiche alors une zone de saisie permettant de définir la mémoire cache à utiliser ainsi que les informations relatives à la mémoire physique (mémoire RAM disponible sur la machine), le cache actuel et le cache après redémarrage (tenant compte de vos modifications) :



La taille de mémoire cache que vous saisissez sera réservée pour la base 4D, quel que soit l'état des ressources de la machine. Ce paramétrage peut être utilisé dans certaines configurations spécifiques, ou lorsque la base est destinée à être utilisée sur des systèmes disparates en termes de mémoire. Dans la plupart des cas, le cache adaptatif est plus performant.

### Mode adaptatif

Dans ce mode, la gestion de la mémoire cache est assurée dynamiquement par le système — dans des limites que vous définissez. Ce principe permet de configurer une mémoire cache performante, adaptée à la plupart des configurations. Pour cela, il vous suffit de cocher l'option **Calcul du cache adaptatif**. 4<sup>e</sup> Dimension affiche alors plusieurs zones de saisie supplémentaires :



La taille du cache est alors calculée dynamiquement en fonction des paramètres définis. Les valeurs proposées par défaut correspondent à une utilisation standard de 4<sup>e</sup> Dimension.

- **Mémoire physique à réserver** : partie de la mémoire RAM à réserver au Système et aux autres applications. Cette valeur peut être augmentée à des fins d'optimisation lorsque d'autres applications consommatrices de mémoire tournent sur le même poste que 4<sup>e</sup> Dimension.
- **Pourcentage de mémoire disponible utilisé pour le cache** : pourcentage de la mémoire restante alloué par défaut au cache.  
Pour obtenir la taille allouée par défaut au cache, il suffit donc d'effectuer le calcul suivant : (Mémoire physique - Mémoire physique à réserver) x Pourcentage de la mémoire utilisé pour le cache.

Dans le mode adaptatif, la taille de mémoire cache varie dynamiquement en fonction des besoins de l'application et du système. Vous pouvez fixer les bornes de ces variations à l'aide des deux options suivantes :

- **Taille maximale** : quantité maximale de mémoire pouvant être utilisée par le cache. Cette valeur ne peut être supérieure à 2000 Mo (2 Go).
- **Taille minimale** : quantité minimale de mémoire devant être réservée pour le cache. Cette valeur ne peut être inférieure à 4 Mo.

*Note* La définition de bornes est utile dans le cadre des bases diffusées sur des machines dont vous ne connaissez pas *a priori* la configuration mémoire. Dans ce cas, les bornes vous permettent de garantir des performances minimales dans tous les cas.

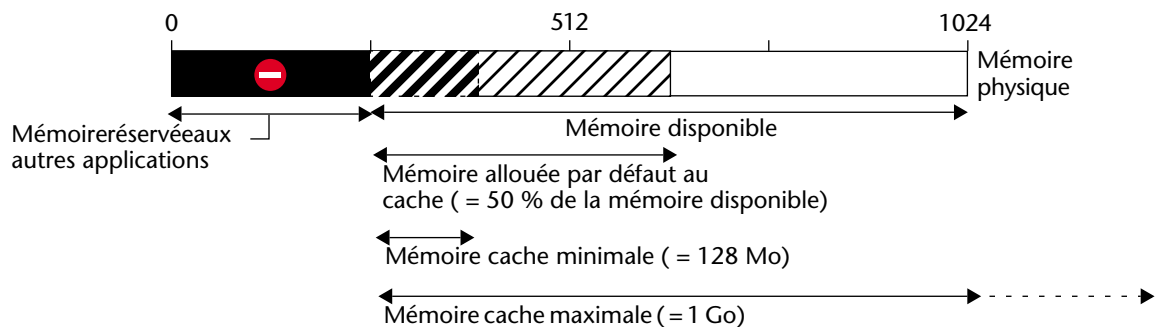
Le schéma suivant illustre ce fonctionnement :

#### Exemple de calcul de la mémoire cache

Mémoire physique à réserver = 256 Mo

Pourcentage de la mémoire disponible utilisé pour le cache = 50 %

Taille maximale = 1 Go Taille minimale = 128 Mo



### Exemples de configurations en mode adaptatif

Les exemples suivants illustrent l'intérêt du cache adaptatif dans différentes configurations logicielles et mémoire.

- Ordinateur avec 768 Mo de RAM faisant tourner un 4D Server unique. Dans ce cas, il est intéressant de lui octroyer le maximum de cache :
  - Mémoire physique à réserver = 256 Mo.
  - Pourcentage de mémoire disponible utilisé pour le cache = 100 % car l'application 4D Server est seule.
  - Taille minimale = 10 Mo et Taille maximale = 1Go (dans cette configuration, ces valeurs sont inutiles).

Taille par défaut du cache :  $(768 - 256) \times 100\% = 512 \text{ Mo}$

- Même configuration que précédemment, mais trois applications 4D Server tournent sur l'ordinateur. Vous souhaitez accorder un cache égal à chaque application. Il suffit de passer la valeur de "Pourcentage de mémoire disponible utilisé pour le cache" de chaque serveur à 33 %.

Taille par défaut du cache pour chaque application :  $(768 - 256) \times 33\% = 170 \text{ Mo}$

- Création d'une application exécutable destinée à une large diffusion dans de multiples configurations. Après tests, il s'avère que l'application fonctionne de manière optimale avec un cache situé entre 20 et 100 Mo.
  - Mémoire physique à réserver = 256 Mo.
  - Pourcentage de mémoire disponible utilisé pour le cache = 50 % afin de laisser de la mémoire aux autres applications.
  - Taille minimale = 20 Mo et Taille maximale = 100 Mo.

Sur une machine disposant de 256 Mo de RAM, 4<sup>e</sup> Dimension utilise le minimum de cache = 20 Mo

Sur une machine disposant de 1 Go de RAM, 4<sup>e</sup> Dimension utilise le maximum de cache = 100 Mo

Sur une machine disposant de 384 Mo de RAM, 4<sup>e</sup> Dimension utilise  $(384 - 256) \times 50\% = 64 \text{ Mo}$

### Reconnexion automatique des clients

Une nouvelle option (apparue dans 4<sup>e</sup> Dimension 2003.3) est disponible dans la page **Configuration** du thème "Client/Serveur" des Préférences : **Utiliser la reconnexion automatique des clients**. Cette fonction bas niveau permet, dans certaines configurations spécifiques,



de reconnecter automatiquement les postes clients en cas de déconnexion inopinée.

Lorsque l'option (globale à tous les clients) est cochée, la fonction de reconnexion est activée automatiquement en cas de perte de la connexion d'un client avec le serveur. Une seule tentative est effectuée : si la reconnexion échoue, l'erreur -10002 est retournée. En cas de succès, une boîte de dialogue d'alerte est affichée et, sous Windows, une icône de notification s'inscrit dans la barre des tâches.

*Note* Pour que le mécanisme de reconnexion automatique puisse être mis en oeuvre, le délai avant déconnexion (*timeout*) côté serveur doit être supérieur à 1 minute.

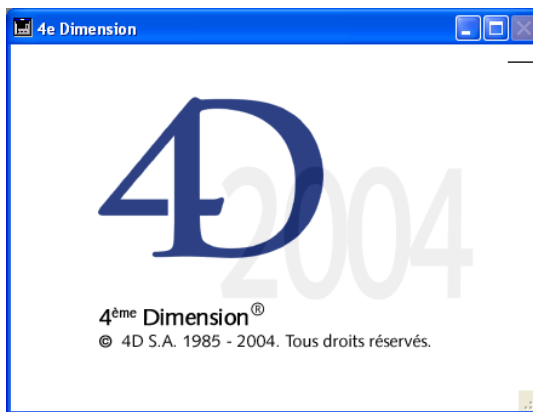
## Nouveautés provenant de 4D Customizer Plus

Le programme utilitaire 4D Customizer Plus n'est plus disponible dans la gamme 4<sup>e</sup> Dimension 2004. Ses fonctions majeures ont été intégrées dans les Préférences de 4<sup>e</sup> Dimension (décrites ci-dessous).

### Afficher fenêtres

La zone "Afficher fenêtres" de la page **Application/Options** des Préférences permet de paramétrer l'affichage par défaut de deux types de fenêtres dans l'application :

- **Affichage de la progression de l'impression** : cette option permet d'activer ou de désactiver l'affichage de la boîte de dialogue de progression d'impression lors d'une impression.
- **Affichage de la fenêtre d'accueil** : lorsque cette option est désélectionnée, la fenêtre d'accueil des modes Utilisation et Menus créés ("splash screen") n'apparaît pas.



Fenêtréd'accueilpar  
défaut du mode  
Menus créés

Lorsque vous cachez cette fenêtre, c'est à vous de gérer par programmation l'affichage de toutes vos fenêtres.

---

*Note* Cette option était disponible dans 4D Customizer Plus via la commande **Fenêtre cachée** du pop-up menu Type de fenêtre.

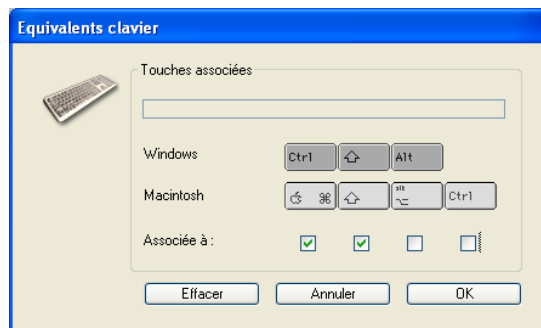
---

### Raccourcis clavier

Cette page du thème "Applications" permet de visualiser et de modifier les raccourcis clavier par défaut de trois opérations élémentaires de 4<sup>e</sup> Dimension : acceptation de saisie, annulation de saisie, ajout d'un sous-enregistrement. Ces raccourcis sont identiques pour chaque plateforme. Les icônes en forme de touches indiquent les correspondances des claviers Windows et MacOS. Les raccourcis par défaut sont les suivants :

- Acceptation de saisie : **Entrée**
- Annulation de saisie : **Echappement**
- Ajout d'un sous-enregistrement : **Ctrl+Maj+ /** (Windows) ou **Commande+Maj+ /** (MacOS)

Pour modifier le raccourci d'une opération, cliquez sur le bouton **Modifier** correspondant. La boîte de dialogue de définition des touches apparaît :



Pour modifier le raccourci, tapez sur votre clavier la nouvelle combinaison à employer et cliquez sur le bouton **OK**.

Si vous souhaitez qu'il n'y ait pas de raccourci clavier associé à une action, cliquez sur le bouton **Effacer**.

### WEDD

La page **Gestion des données** du thème "Base de données" contient la zone WEDD, permettant de définir une signature WEDD.

Une signature WEDD associe un fichier de structure à un fichier de données. Un fichier de structure contenant une signature WEDD ne pourra être ouvert qu'avec un fichier de données contenant la même signature WEDD, et *vice versa*. Ainsi, vous pouvez interdire l'utilisation de fichiers de données incompatibles. Vous pouvez également définir des signatures WEDD si vous travaillez avec plusieurs bases de données et voulez éviter d'ouvrir accidentellement un fichier de données qui n'appartient pas à une structure particulière.

Par défaut, ce paramètre n'est pas défini dans les bases de données. Pour ajouter une signature WEDD à une base, saisissez simplement une valeur personnalisée dans la zone WEDD. Elle peut être composée de n'importe quelle combinaison de caractères. Le fichier de structure et le fichier de données courant sont alors liés.

Pour supprimer une signature WEDD, il suffit d'effacer le contenu de cette zone.

## Script Manager

La page **Script Manager** du thème "Base de données" permet de modifier certains paramètres lors de l'utilisation de la base sous Script Manager. Le Script Manager est la partie du système qui gère l'écriture des langues à caractères non romains (japonais, arabe...). Si vous utilisez 4<sup>e</sup> Dimension sous Script Manager, vous pouvez avoir besoin de modifier certains paramètres présents dans cette fenêtre.

- **Code ASCII du 0 (zéro)** : code ASCII du caractère zéro. Sur un système français ou américain, sa valeur est de 48. Il peut être différent sur un autre système (arabe par exemple).
- **... et suivant** : valeur de l'octet qui suit le zéro. Ce paramètre n'est utilisé que si vous localisez en système arabe ou en hébreu. Dans tous les autres cas, conservez ce paramètre à 0 (zéro).
- **Comparaisons** : ce menu déroulant propose quatre choix possibles pour vos tris et recherches :
  - *4<sup>e</sup> Dimension* : utilise la ressource TRIC ou, à défaut, la ressource TRI#.
  - *Système* : utilise les routines de comparaison et de mise en majuscules de la ROM (ignore la ressource TRIC).
  - *Allemand pour V2.2* : méthode de comparaison utilisée en allemand avec des dispositions spéciales pour le caractère "ß".
  - *Turc* : méthode de comparaison comportant des dispositions spéciales pour les caractères turcs.

- **Ressources TRIC (nouvelle base de données) et Ressources TRIC (base de données ouverte)** : par défaut, les tris sont basés sur une table universelle proposée par 4<sup>e</sup> Dimension (“e”, “é”, “è”, “ë”, “ê” sont triés séparément ; en revanche, la recherche de “é” trouvera “e”). Cette table peut différer selon les langues (caractères spéciaux...).

Vous pouvez modifier la ressource TRIC en fonction de la langue du système. Vous disposez de deux menus, correspondant à deux usages différents :

- le menu “nouvelle base de données” permet de modifier la ressource TRIC pour toutes les bases de données créées avec 4<sup>e</sup> Dimension. Utilisez ce menu si vous développez couramment des bases à destination d’un même système.
- le menu “base de données ouverte” permet de modifier la ressource TRIC uniquement pour la base de données courante. Utilisez ce menu si vous avez développé ponctuellement une base destinée à être utilisée sur un système spécifique.

---

*Note* La modification des ressources TRIC entraîne la reconstruction des tables de tris de 4D et la réindexation automatique de la base.

---

- **Calcul sur dates** : cette option a pour but d’ajuster le paramétrage de 4<sup>e</sup> Dimension pour les calculs sur les dates. Dans le cas du calendrier *Farsi*, le codage doit se faire sur 4 octets. Tous les autres cas nécessitent l’option par défaut (8 octets).

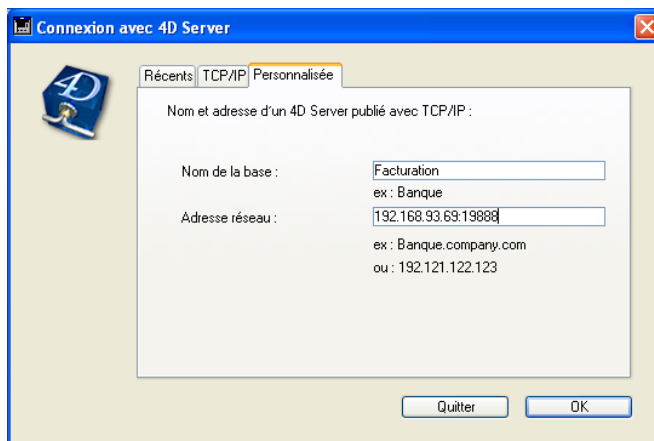
### Numéro de port

Cette option de la page **Configuration** du thème “Client/Serveur” permet de modifier le numéro de port TCP sur lequel 4D Server publie la base de données. Cette information est stockée dans la structure de la base et sur chaque poste client. Par défaut, le numéro de port TCP utilisé par 4D Server et 4D Client est le 19813.

La personnalisation de cette valeur est nécessaire lorsque vous souhaitez utiliser plusieurs applications 4D sur la même machine avec le protocole TCP ; dans ce cas, vous devez spécifier un numéro de port différent pour chaque application.

Lorsque vous modifiez cette valeur depuis 4D Server ou 4D Client, elle est automatiquement répercutée sur tous les postes 4D Client connectés à la base. Pour mettre à jour les autres postes clients non connectés, il suffira, lors de leur connexion suivante, de saisir le nouveau numéro de port (précédé de deux-points) derrière l’adresse IP

du poste serveur dans la page **Personnalisée** de la boîte de dialogue de connexion. Par exemple, si le nouveau numéro de port est le 19888 :



*Note* Seules les bases publiées sur le même port que celui défini dans 4D Client sont visibles dans la page de publication dynamique **TCP/IP**.

## Publication

La page **Publication** du thème "Client-Serveur" comporte plusieurs nouvelles options liées à la publication de la base par 4D Server.

- **Publier** : cette option permet d'indiquer si la base 4D Server doit apparaître ou non dans la liste des bases publiées.
  - lorsque l'option est cochée, la base est rendue publique, elle apparaît dans la liste des bases publiées (option par défaut).
  - lorsque l'option est désélectionnée, la base n'est pas rendue publique, elle n'apparaît pas dans la liste des bases publiées (page **TCP/IP**). Pour se connecter, les utilisateurs doivent saisir manuellement l'adresse de la base dans la page **Personnalisée** de la boîte de dialogue de connexion.
- **Table de configuration Accepter/Refuser**  
 Cette table vous permet de définir des règles de contrôle d'accès à la base en fonction de l'adresse IP des postes 4D Client. Cette option permet de renforcer la sécurité par exemple pour des applications stratégiques.

*Note* Cette table de configuration ne contrôle pas les connexions Web.

Le fonctionnement de la table de configuration est le suivant :

- La colonne “Autoriser-Refuser” permet de sélectionner le type de règle à appliquer (**Autoriser** ou **Refuser**) à l’aide d’un pop up menu. Pour ajouter une règle d’adresses, cliquez sur le bouton **Ajouter**. Une nouvelle ligne apparaît dans la table. Le bouton **Supprimer** permet de supprimer la ligne courante.
- La colonne “Adresse IP” permet de désigner la ou les adresse(s) IP concernées par la règle. Pour spécifier une adresse, cliquez dans la colonne et saisissez l’adresse sous la forme 123.45.67.89. Vous pouvez utiliser le caractère \* (étoile) pour spécifier des adresses du type “commence par”. Par exemple, 192.168.\* indique toutes les adresses débutant par 192.168.
- L’application des règles s’effectue dans l’ordre d’affichage de la table. Si deux règles sont contradictoires, la priorité sera accordée à la règle située le plus haut dans le tableau. Vous pouvez réordonner les lignes en modifiant le tri courant (cliquez sur un en-tête de colonne pour alterner le sens de tri). Vous pouvez également déplacer des lignes par glisser-déposer.
- Pour des raisons de sécurité, seules les adresses correspondant à une règle d’autorisation explicite pourront se connecter. En particulier, si la table contient uniquement une ou plusieurs règle(s) de type **Refuser**, toutes les adresses seront refusées car aucune ne satisfera à au moins une règle. Si vous souhaitez refuser certaines adresses et autoriser toutes les autres, ajoutez une règle **Autoriser \*** à la fin de la table. Par exemple :
  - Refuser 192.168.\* (*refuser toutes adresses débutant par 192.168*)
  - Autoriser \* (*et autoriser les autres*)

Par défaut, aucune restriction de connexion n’est appliquée par 4D Server : la première ligne de la table contient le libellé **Autoriser** et le caractère \* (toute adresse).

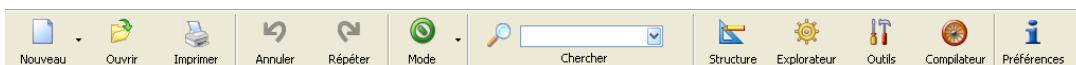
# Interface

L'interface de la plupart des éditeurs du mode Structure de 4<sup>e</sup> Dimension a été modifiée.

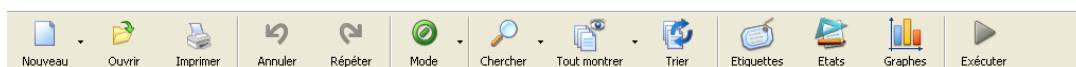
## Barres d'outils

Les nouvelles barres d'outils (mode Structure et mode Utilisation), plus fonctionnelles, permettent un accès direct aux principales fonctions :

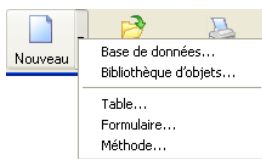
### Mode Structure



### Mode Utilisation

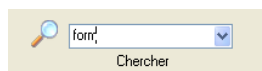


- Les boutons comportant une icône en forme de triangle sont associés à des menus :



Ces boutons fonctionnent de la manière suivante : un clic sur le bouton exécute l'action courante associée. Cette action est la dernière sélectionnée dans le menu. Pour afficher le menu et sélectionner une autre action, cliquez dans la partie droite du bouton.

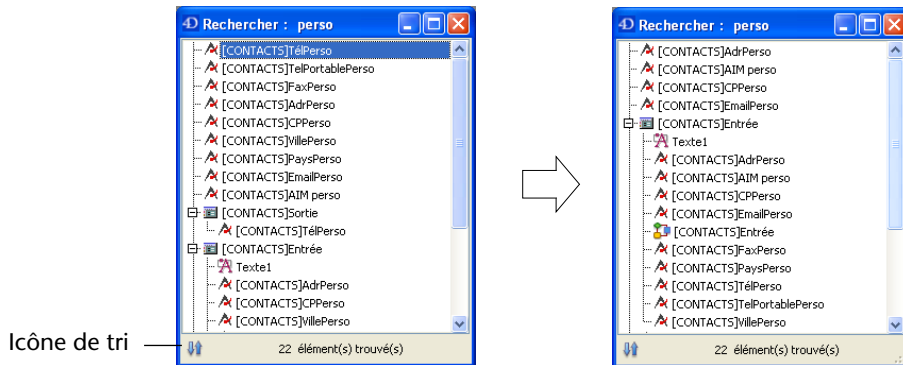
- La zone de recherche rapide permet de lancer une recherche standard dans la base. Pour lancer directement une recherche, inscrivez la chaîne à rechercher dans la zone de saisie puis appuyez sur la touche **Retour chariot**.



La recherche est effectuée sans options (du type "contient" dans tous les objets de la base).

Si vous souhaitez effectuer une recherche avec options, vous pouvez cliquer sur l'icône en forme de loupe afin d'afficher la boîte de dialogue standard de recherche.

A noter que la fenêtre de résultat de la recherche comporte une nouvelle icône permettant de trier le contenu de la fenêtre (listes et éléments) par ordre alphabétique :



## Réorganisation des menus

Les barres de menus de 4<sup>e</sup> Dimension (mode Utilisation et mode Structure) ont été réorganisées afin de proposer une logique plus standard. Cette réorganisation permet un accès plus intuitif aux diverses fonctions de l'application.

Ce paragraphe décrit les modifications majeures effectuées dans les menus, c'est-à-dire :

- en Mode Structure, la modification des menus **Fichier** et **Edition** et l'ajout des nouveaux menus **Exécuter** et **Fenêtres**,
- en Mode Utilisation, la modification des menus **Fichier** et **Edition** et l'ajout des nouveaux menus **Enregistrements** et **Outils**.

Les modifications des menus associés aux éditeurs sont décrits dans la documentation de ces éditeurs.

## Mode Structure

Le tableau suivant décrit les principales nouvelles commandes et commandes modifiées dans le mode Structure de 4<sup>e</sup> Dimension 2004 et fournit leur équivalent dans la version précédente de 4<sup>e</sup> Dimension.

Commandes modifiées en Mode Structure	Equivalent 4 <sup>e</sup> Dimension 2003
Fichier	
Nouveau> Base	Fichier/Ouvrir une base
Nouveau> Bibliothèque d'objets	-
Nouveau> Table	Structure/Nouvelle table (Editeur de structure)
Nouveau> Formulaire	Configuration/Nouveau formulaire
Nouveau> Méthode	Configuration/Nouvelle méthode



Ouvrir> Base	Fichier/Ouvrir une base
Ouvrir> Bibliothèque d'objets	-
Ouvrir> Formulaire	Configuration/Modifier formulaire
Ouvrir> Méthode	Configuration/Modifier méthode
Edition	
Chercher> <i>Commandes de recherche</i>	Edition/ <i>Commandes de recherche</i>
Exécuter (menu identique en mode Structure et Utilisation)	
En compilé / En interprété	Mode/Exécuter en compilé Mode/Exécuter en interprété
Méthode	<i>Mode Utilisation/Sécial/Exécuter une méthode</i>
Explorateur d'exécution	Outils/Explorateur d'exécution
Lancer le serveur Web	<i>Mode Utilisation/Web/Lancer le serveur Web</i>
Tester le serveur Web	-
Structure	
Explorateur> <i>Pages de l'Explorateur</i>	Outils/Explorateur
Boîte à outils> Utilisateurs	Outils/Mots de passe + <i>menu Mot de passe</i>
Boîte à outils> Groupes d'utilisateurs	Outils/Mots de passe + <i>menu Mot de passe</i>
Boîte à outils> Menus	Outils/Editeur de barres de menus
Boîte à outils> Bibliothèque d'images	Outils/Bibliothèque d'images
Boîte à outils> Messages d'aide	Palette des propriétés d'objets
Boîte à outils> Enumérations	Outils/Editeur d'énumérations
Boîte à outils> Feuilles de style	Edition/Préférences, page Feuille de style
Boîte à outils> Filtres	Edition/Préférences, page Formatage
Boîte à outils> Boîte à outils	-
Structure	Outils/Structure
Assistant Web Services	Outils/Assistant Web Services
Compilateur	Outils/Compilateur
Générer l'application	Fichier/Générer l'application
Fenêtre (organisation et liste des fenêtres ouvertes) -	

**Mode Utilisation** Le tableau suivant décrit les principales nouvelles commandes et commandes modifiées dans le mode Utilisation de 4<sup>e</sup> Dimension 2004 et fournit leur équivalent dans la version précédente de 4<sup>e</sup> Dimension.

Commandes modifiées en Mode Utilisation	Equivalent 4 <sup>e</sup> Dimension 2003
<b>Fichier</b>	
Nouveau> Base	Fichier/Ouvrir une base
Nouveau> Fichier de données	Clic sur Autre... dans la page Options de la boîte de dialogue d'ouverture
Ouvrir> Base	Fichier/Ouvrir une base
Ouvrir> Fichier de données	Clic sur Autre... dans la page Options de la boîte de dialogue d'ouverture
Importer> Du fichier	Fichier/Importer des données
Importer> De la source ODBC	-
Exporter> Vers fichier	Fichier/Exporter des données
Exporter> Vers la source ODBC	-
Vider le cache	Ctrl+w / Commande+w
Segments de données	<i>Mode Structure/Outils/Structure</i> puis <i>Structure/Segments de données</i>
Sauvegarder	-
Vérifier fichier d'historique	-
<b>Edition</b>	
Préférences (Windows)	<i>Mode Structure/Edition/Préférences</i>
<b>Exécuter (menu identique en mode Structure et Utilisation)</b>	
En compilé / En interprété	Mode/Exécuter en compilé Mode/Exécuter en interprété
Méthode	Spécial/Exécuter une méthode
Explorateur d'exécution	Outils/Explorateur d'exécution
Lancer le serveur Web	Web/Lancer le serveur Web
Arrêter le serveur Web	Web/Arrêter le serveur Web
Tester le serveur Web	-
<b>Enregistrements</b>	
Nouvel enregistrement en liste	Saisie/Saisie en liste + Saisie/Nouvel enregistrement
Nouvel enregistrement	Saisie/Nouvel enregistrement
Modifier l'enregistrement	Saisie/Modifier l'enregistrement
Tout montrer	Sélection/Tout montrer

Sous sélection	Sélection/Sous sélection
Recherche> <i>Commandes de recherche</i>	Sélection/ <i>Commandes de recherche</i>
Trier	Sélection/Trier
Liste des tables	Fichier/Choisir table/formulaire <i>ou</i> raccourci Ctrl (Commande)+barre d'espace
Appliquer une formule	Saisie/Appliquer une formule
Outils	
Etats rapides	Etats/Rapides
Etiquettes	Etats/Etiquettes
Graphes	Etats/Graphes
4D Chart	Plug-ins/4D Chart
OLE tools	Plug-ins/OLE tools
<i>Autres plug-ins</i>	Plug-ins/ <i>Autres plug-ins</i>
Fenêtre (organisation et liste des fenêtres ouvertes) -	

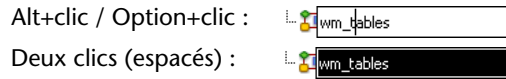
- Note* Différentes fonctions existantes dans le mode Utilisation de 4<sup>e</sup> Dimension 2003 ont été remplacées ou déplacées :
- Les commandes de gestion du fichier d'historique (menu **Fichier**) sont désormais prises en charge dans le thème "Sauvegarde" des Préférences de l'application.
  - La commande **Saisie en liste** (menu **Saisie**) a été supprimée car les nouveaux mécanismes de gestion des listes la rendent inutile (cf. [paragraphe "Gestion des listes d'enregistrements", page 165](#)).
  - Les commandes d'édition des filtres d'exportation et d'importation (menu **Spécial**) sont désormais disponibles dans les Préférences de l'application).

## Raccourcis clavier

Les raccourcis-clavier ont été réorganisés dans 4<sup>e</sup> Dimension 2004 et certains raccourcis redondants ont été supprimés.

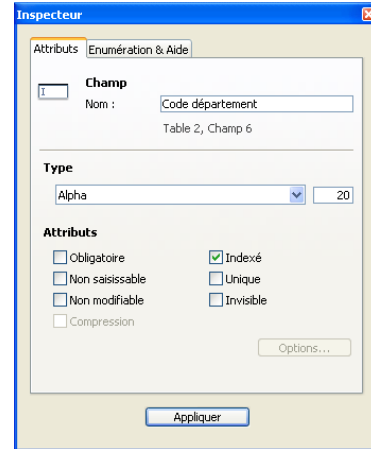
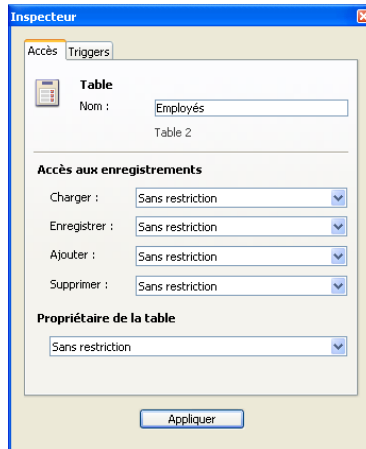
- Le raccourci **Ctrl+Barre d'espace** (Windows) ou **Commande+Barre d'espace** (MacOS) permettant d'afficher la Liste des tables en mode Utilisation a été remplacé par la commande **Liste des tables** du menu **Enregistrements** et son équivalent **Ctrl+0** (Windows) / **Commande+0** (MacOS).
- Le raccourci **Ctrl+Barre d'espace** (Windows) ou **Commande+Barre d'espace** (MacOS) permettant d'afficher l'Explorateur en mode Structure a été remplacé par l'équivalent **Ctrl+,** (Windows) / **Commande+,** (MacOS).

- Le raccourci **Ctrl+w** (Windows) ou **Commande+w** (MacOS) permettant de forcer l'écriture du cache en mode Utilisation a été remplacé par la commande **Ecrire le cache** du menu **Fichier** et son équivalent **Alt+Ctrl+Maj+s** (Windows) ou **Option+Commande+Maj+s** (MacOS).
- Le raccourci **Ctrl+clik** (Windows) ou **Commande+clik** (MacOS) permettant d'insérer le curseur d'édition dans un élément de liste (par exemple un nom de méthode dans l'Explorateur ou un menu dans l'éditeur de menus) a été remplacé par le raccourci **Alt+clik** (Windows) ou **Option+clik** (MacOS).  
A noter que **cliquer deux fois** sur un élément de liste permet également de le passer en édition (dans ce cas l'élément est entièrement sélectionné) :



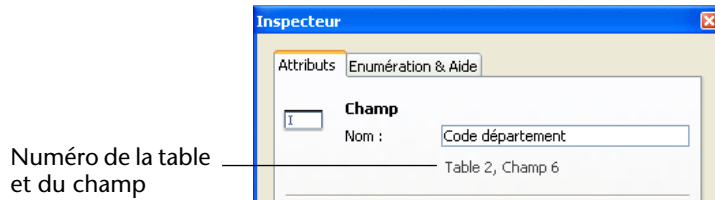
## Inspecteur

L'accès aux propriétés des tables, des champs et des liens dans la fenêtre de Structure s'effectue désormais via une palette unique nommée **Inspecteur** :



Le contenu de la palette est dynamiquement mis à jour en fonction de l'objet sélectionné.

A noter que la palette affiche les numéros des tables et des champs :



*Note* Il n'est pas possible d'appliquer une couleur à l'objet sélectionné via la palette de l'Inspecteur. Cette propriété est désormais gérée uniquement à l'aide du menu contextuel.

## Explorateur d'exécution

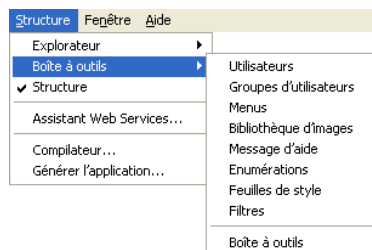
Le contenu de la page **Evaluation** de l'Explorateur d'exécution a été modifié :

- pour plus de clarté, les éléments auparavant affichés dans la liste **Informations** ont été réorganisés dans deux listes séparées, **Informations** et **Web**.
- des informations supplémentaires sont disponibles lorsque vous sélectionnez la commande **Activer le suivi d'activité** dans le menu contextuel de la page. Attention, cette activation ralentit les traitements. Ces informations, regroupées dans les listes **Séquenceur** et **Réseau**, permettent un contrôle avancé de l'activité interne de l'application.

## Boîte à outils

Les différents éditeurs secondaires de 4<sup>e</sup> Dimension ont été regroupés au sein d'un même environnement : la boîte à outils.

La boîte à outils est accessible via la commande hiérarchique **Boîte à outils** dans le menu **Structure** :

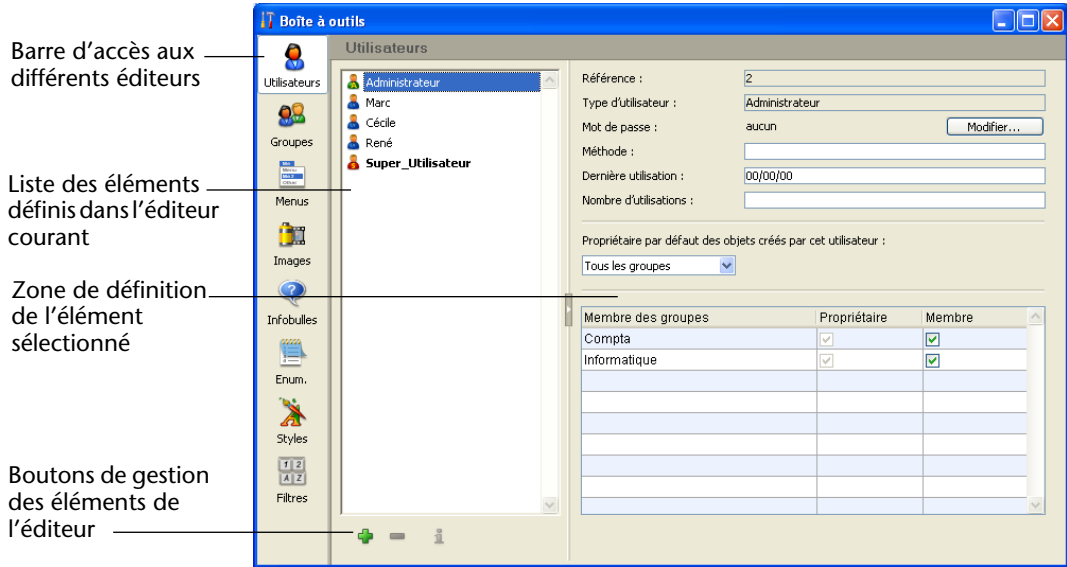


Chaque sous-commande provoque l'affichage de la boîte à outils sur la page correspondante.



La commande **Boîte à outils** ouvre la boîte à outils à la page courante. Vous pouvez également utiliser le bouton **Outils** de la barre d'outils.


Par défaut, la boîte à outils affiche la première page (Utilisateurs). Elle se compose des éléments suivants :



Les icônes des éditeurs Utilisateurs et Groupes correspondent à celles utilisées dans le système. Elles apparaissent ainsi sous MacOS X :

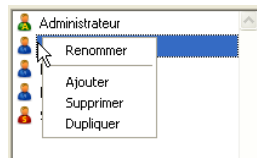


- Les boutons situés au bas de la liste permettent d'ajouter ou de supprimer des éléments en fonction de l'éditeur affiché : utilisateurs, groupes, barres de menus, images, messages, énumérations, styles et filtres/formats.

Ajouter un élément —  — Supprimer l'élément sélectionné

Des boutons supplémentaires proposant des fonctions spécifiques sont présents dans certains éditeurs.

- La liste des éléments comporte également un menu contextuel (affiché via un **clic droit** sous Windows ou **Control+clic** sous MacOS), donnant accès aux fonctions d'édition des éléments (ajout, suppression, etc.) :



Les fonctions présentes dans les éditeurs de la boîte à outils sont généralement identiques aux fonctions présentes dans les versions précédentes de 4<sup>e</sup> Dimension, à quelques exceptions près (décrites dans les pages suivantes).

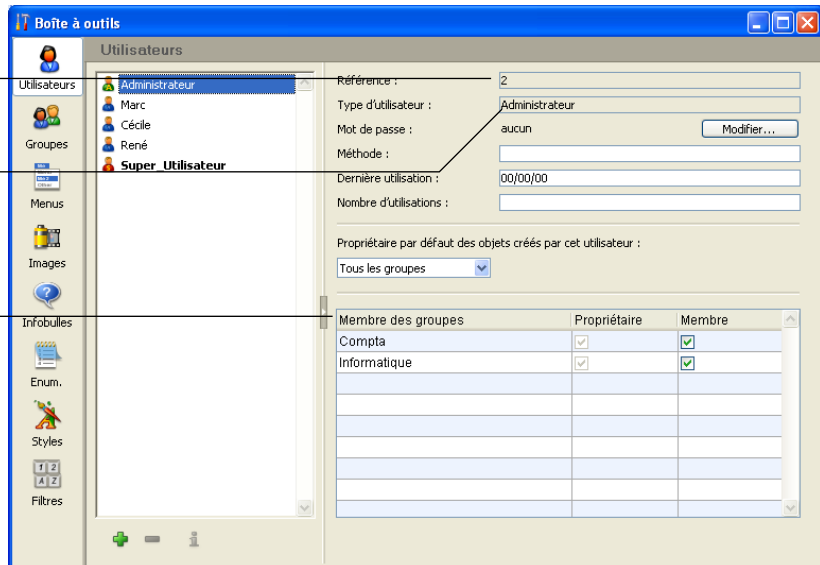
## Utilisateurs

Cet éditeur permet de définir les profils des utilisateurs de la base.

Numéro de référence de l'utilisateur

Type d'utilisateur

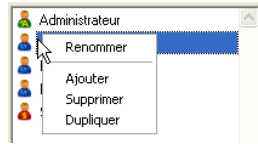
Zone de définition de l'appartenance aux groupes



- La liste des utilisateurs affiche tous les utilisateurs “visibles” par l'utilisateur courant, c'est-à-dire :
  - pour le Super\_Utilisateur : tous les utilisateurs,
  - pour l'Administrateur : les utilisateurs qu'il a créés (icônes vertes).

Les icônes de ces deux utilisateurs affichent respectivement les lettres **S** et **A**. Il est possible de renommer ces deux utilisateurs.

Le menu contextuel de la liste permet de **Renommer**, **Ajouter**, **Supprimer** ou **Dupliquer** un utilisateur :



*Note* Pour renommer un utilisateur, vous pouvez également cliquer deux fois sur un nom, il passe alors en édition.

---

- Le champ “Référence” indique le numéro de référence de l'utilisateur sélectionné. Ce numéro est utilisé par les commandes du langage.
- Le champ “Type d'utilisateur” indique la provenance du profil d'utilisateur. Les types d'utilisateurs sont les suivants :
  - Super\_Utilisateur : utilisateur Super\_Utilisateur
  - Administrateur : utilisateur Administrateur
  - Développeur : utilisateur créé par le Super\_Utilisateur
  - Utilisateur : utilisateur créé par l'Administrateur
- Le tableau “Membre des groupes” affiche les groupes définis dans la base :
  - vous pouvez ajouter l'utilisateur sélectionné à un groupe en cochant l'option correspondante dans la colonne Membre. Vous pouvez également extraire l'utilisateur du groupe en désélectionnant l'option.
  - la colonne Propriétaire indique si l'utilisateur sélectionné est propriétaire du groupe. Cette colonne n'est pas modifiable.

L'appartenance des utilisateurs aux groupes peut également être définie par groupe dans la page **Groupes**.



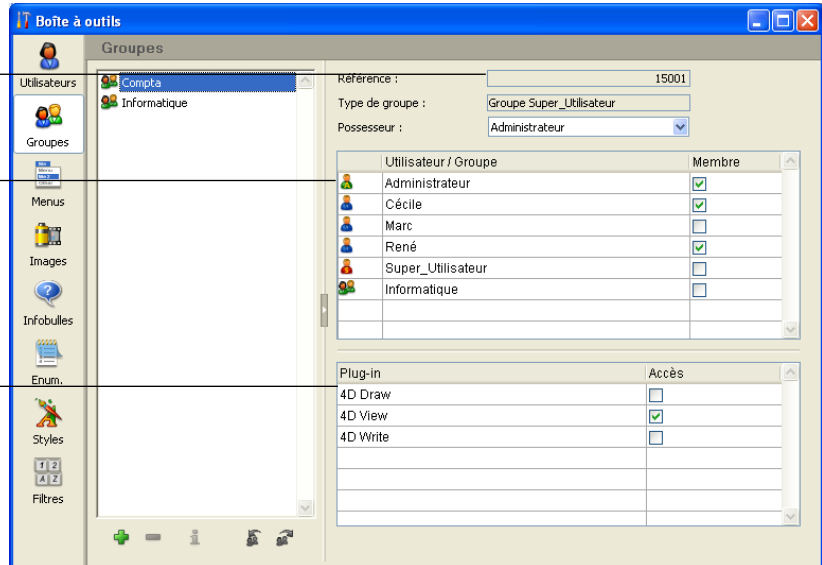
## Groupes

Cet éditeur permet de définir les groupes d'utilisateurs de la base.

Numéro de référence du groupe

Zone de définition de l'appartenance des utilisateurs

Zone de définition de la répartition des accès aux plug-ins

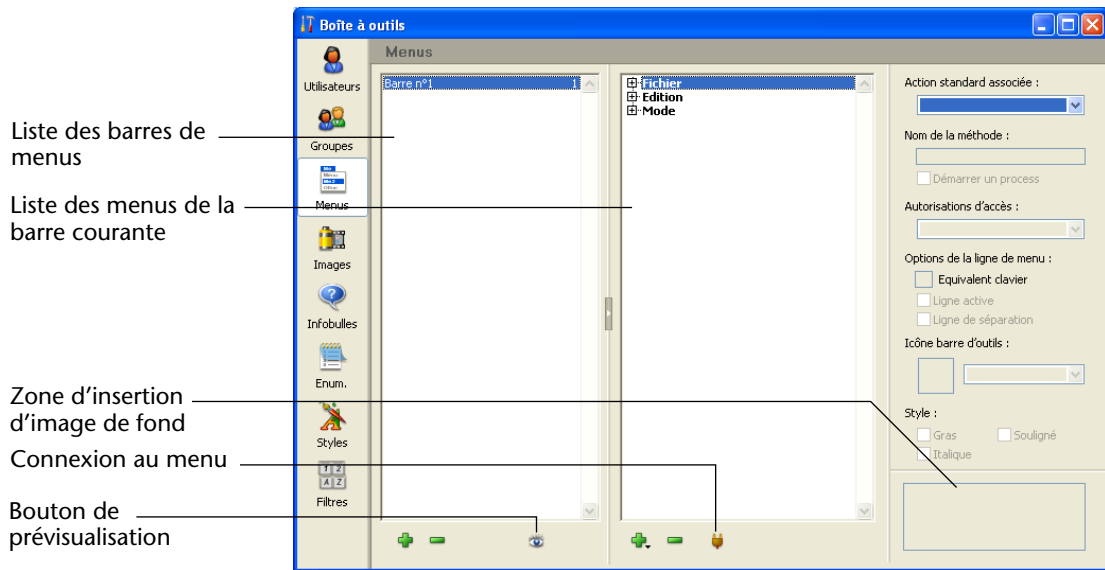



- Le champ "Référence" indique le numéro de référence du groupe sélectionné. Ce numéro est utilisé par les commandes du langage.
- Le champ "Type de groupe" indique la provenance du groupe. Les types de groupes sont les suivants :
  - Groupe Super\_Utilisateur : groupe créé par le Super\_Utilisateur
  - Groupe Administrateur : groupe créé par l'Administrateur
- Le tableau des membres affiche tous les utilisateurs et les groupes définis dans la base, à l'exception du groupe sélectionné. L'option de la colonne "Membre" désigne les utilisateurs et les groupes appartenant au groupe sélectionné.  
Vous pouvez ajouter ou supprimer des utilisateurs et des groupes du groupe sélectionné en cliquant sur cette option.  
L'appartenance des utilisateurs aux groupes peut également être définie par utilisateur dans la page **Utilisateurs**.
- Le tableau "Plug-ins" permet de répartir les accès des membres du groupe sélectionné pour chaque plug-in.
- Deux boutons supplémentaires permettent d'importer et d'exporter les utilisateurs sous forme de fichier sur disque (seul l'Administrateur de la base peut utiliser cette fonction) :

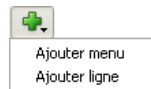


### Menus

Cet éditeur permet de définir les barres de menus de la base.

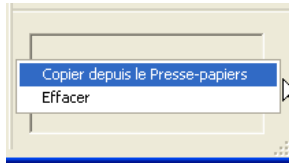


- Le bouton de prévisualisation  permet de passer en mode prévisualisation dans lequel la barre de menus, les menus ainsi que la fenêtre d'accueil sont affichés.
- Le bouton d'ajout situé au-dessous de la liste des menus est associé à un pop up menu permettant d'ajouter un menu ou une ligne de menu :



- Pour connecter un menu au menu sélectionné, cliquez sur l'icône supplémentaire située au-dessous de la liste des menus, puis choisissez un menu dans la liste qui apparaît.
- Zone d'insertion d'image de fond : cette zone permet d'associer une image de fond personnalisée à la barre de menus sélectionnée. Cette image sera utilisée à la place du "4D" bleu affiché par défaut dans la fenêtre d'accueil du mode Menus créés.

Pour associer une image de fond personnalisée à la barre de menus, il suffit de placer l'image dans le Presse-papiers puis de cliquer dans la zone. Un pop up menu apparaît, permettant de coller l'image du Presse-papiers :



La commande **Effacer** permet de supprimer l'image personnalisée et de rétablir l'image par défaut.

*Note* Il n'est plus possible de coller une image en mode prévisualisation.

## Images

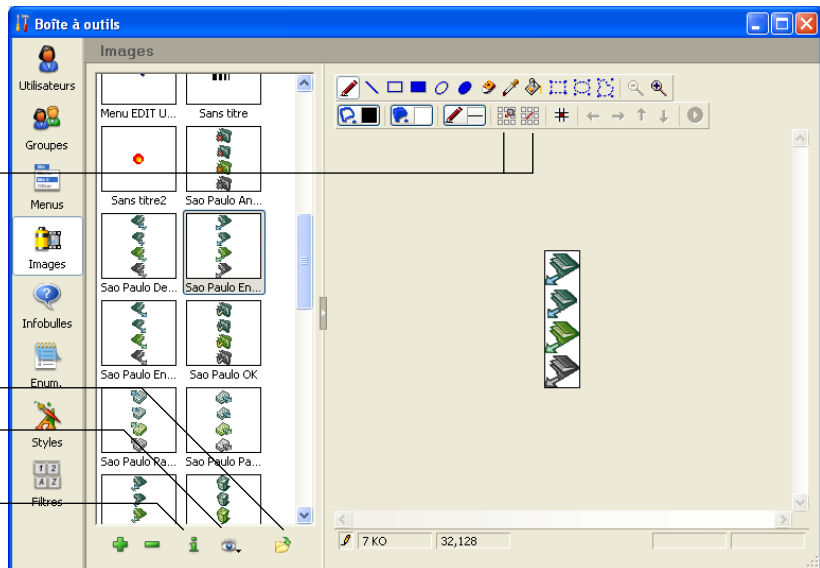
La bibliothèque d'images permet de gérer les différentes images utilisées dans la base de données.

Accès à l'assistant de gestion d'images

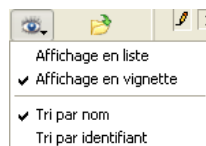
Import d'image

Gestion de la liste d'images



Affichage des propriétés de l'image



- L'accès aux fonctions de gestion de la liste des images s'effectue via le pop up menu situé au-dessous de la liste :

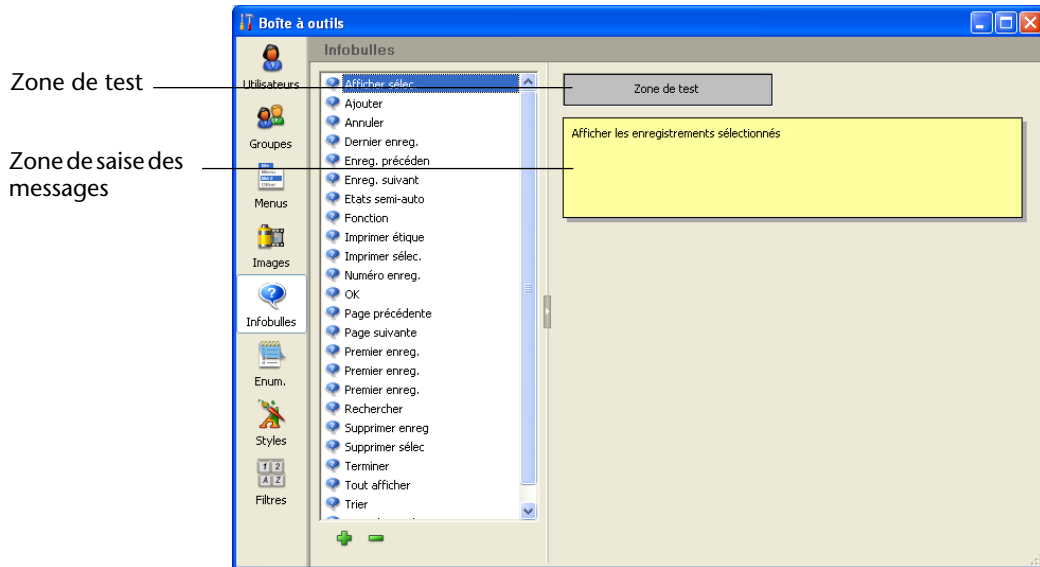


A noter qu'il est désormais possible de trier les images par nom ou par numéro.

- Pour importer une image, il suffit de cliquer sur le bouton .
- L'assistant de gestion d'images (insertion ou suppression) est accessible via les boutons  situés dans la barre d'outils de la fenêtre.

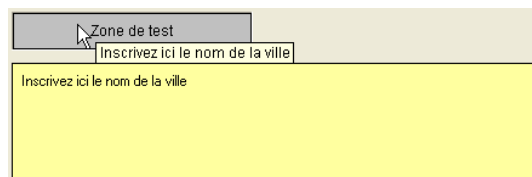
### Infobulles

Ce nouvel éditeur permet de créer et de gérer les infobulles (messages d'aide) utilisées dans les formulaires de l'application :



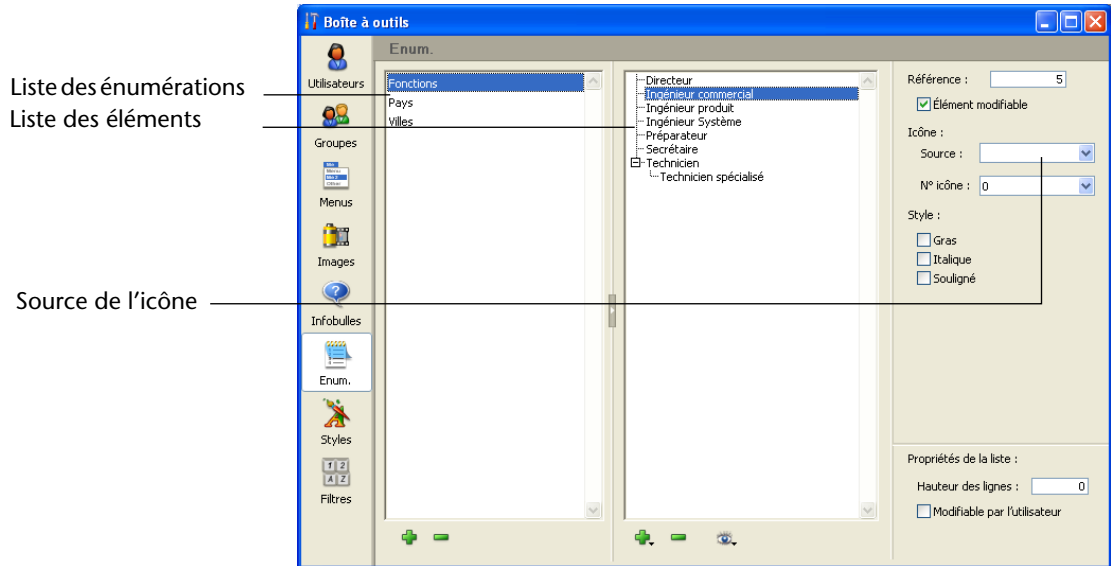
*Note* Dans les versions précédentes de 4<sup>e</sup> Dimension, cet éditeur était situé dans la palette Propriétés de l'objet.

- Le menu contextuel de la liste des messages permet de **Renommer**, **Ajouter**, **Supprimer** ou **Dupliquer** un message.
- La zone de test permet de visualiser l'affichage du message d'aide tel qu'il apparaîtra dans la base. Pour l'activer, sélectionnez le message à tester et placez le curseur de la souris au-dessus de la zone de test, sans cliquer :



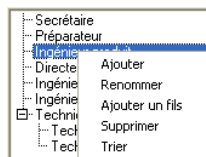
## Énumérations

Cet éditeur permet de définir les énumérations utilisées dans la base.

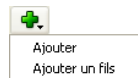


Les fonctions proposées par cet éditeur sont globalement identiques à celles de l'éditeur d'énumérations présent dans les versions précédentes de 4<sup>e</sup> Dimension.

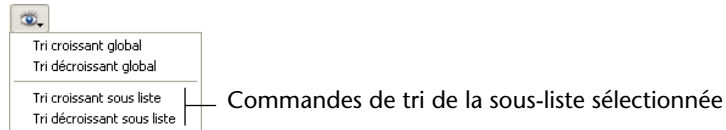
- Le menu contextuel de la liste des énumérations permet de **Renommer**, **Ajouter**, **Supprimer** ou **Dupliquer** une énumération.
- Le menu contextuel de la liste des éléments comporte en outre la commande **Ajouter un fils** :



Les commandes d'ajout d'éléments se retrouvent également dans le menu associé à l'icône d'insertion :



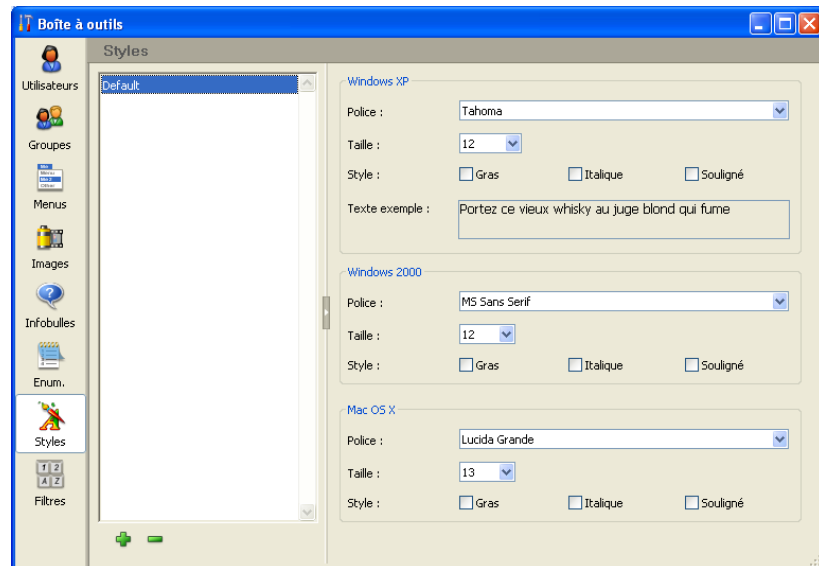
- Il est désormais possible, dans le cadre d'une liste hiérarchique, de trier uniquement une sous-liste:



- A noter enfin la possibilité de définir la provenance de l'icône associée à chaque élément via le menu **Source** : **Ressource 'cicn'**, **Ressource 'pict'** ou **Bibliothèque d'images**. Le numéro d'icône saisi tiendra compte de cette information.

### Styles

Cet éditeur permet de créer et de gérer les feuilles de style utilisées dans l'application.



Le menu contextuel de la liste des styles permet de **Renommer**, **Ajouter**, **Supprimer** ou **Dupliquer** un style.

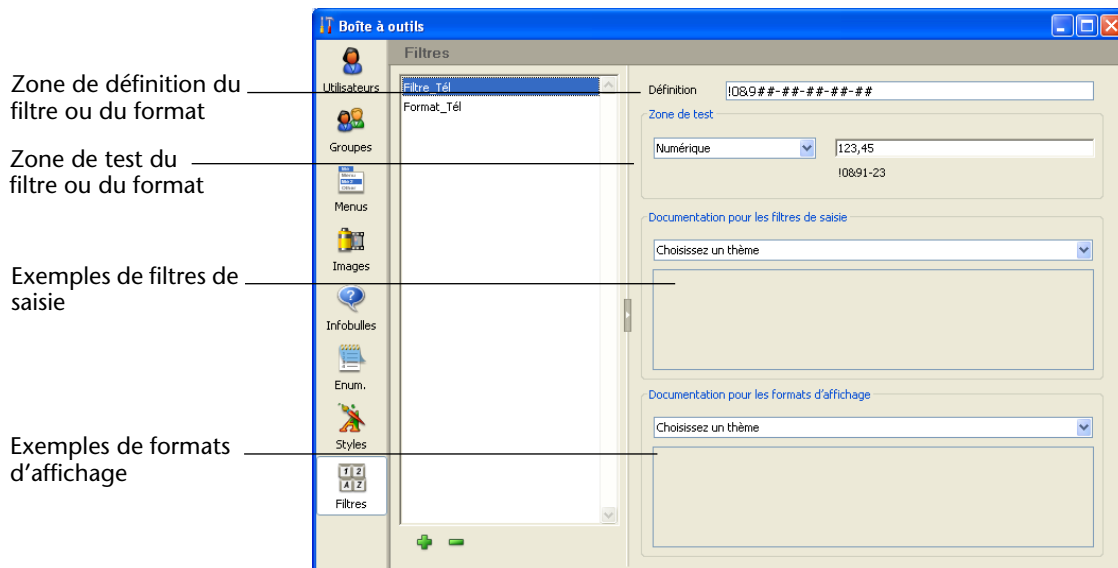
Conformément aux nouveaux principes relatifs à la prise en charge des interfaces dans 4<sup>e</sup> Dimension 2004 (cf. [paragraphe "Interface de plate-forme"](#), page 148), chaque feuille de style peut désormais être déclinée en trois plates-formes : **Windows XP**, **Windows 2000** et **Mac OS X**.

Chaque plate-forme est définie dans une zone spécifique de l'éditeur. L'ordre dans lequel sont affichées les zones dépend de la plate-forme d'exploitation courante de 4<sup>e</sup> Dimension : par exemple, la zone **MacOS X** (ainsi que le texte exemple associé) est affichée en haut de la page lorsque 4<sup>e</sup> Dimension est exécuté sous MacOS X.

Les autres fonctions de l'éditeur sont identiques à celles de l'éditeur des versions précédentes de 4<sup>e</sup> Dimension.

## Filtres

Cet éditeur permet de créer et de gérer des filtres de saisie et des formats d'affichage personnalisés.

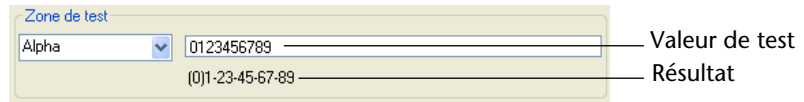


Cet éditeur contient de nouveaux outils destinés à faciliter la définition des formats d'affichage et des filtres de saisie personnalisés.

- Le menu contextuel de la liste des filtres et formats permet de **Renommer**, **Ajouter**, **Supprimer** ou **Dupliquer** un filtre ou un format.
- **Définition** : zone de saisie de la définition du filtre/format. Il est possible de placer des caractères dans cette zone à l'aide d'un double-clic depuis la zone d'exemples située en bas de la fenêtre.
- **Zone de test** : cette zone permet de contrôler le fonctionnement du filtre/format que vous êtes en train de définir. Dans un premier temps, vous devez désigner dans le menu associé le type de données auxquelles le filtre/format sera appliqué : Alpha, Numérique, Date ou Heure.

Une fois ce paramètre défini, saisissez une valeur de test dans la zone associée :

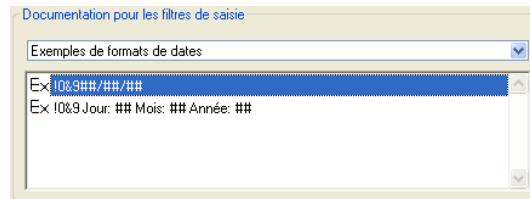
- pour les filtres de saisie, le fonctionnement du filtre est contrôlé au cours de la saisie,
- pour les formats d'affichage, appuyez sur la touche **Entrée** ou **Retour chariot** à l'issue de la saisie. Le résultat de l'application du format apparaît sous la zone de saisie :



- **Documentation pour les filtres de saisie / Documentation pour les formats d'affichage** : ces zones sont composées d'un menu et d'une zone d'affichage.

Le menu permet de désigner un thème sur lequel vous souhaitez obtenir des informations ou des exemples.

Lorsqu'un thème est sélectionné, les caractères et les informations correspondants apparaissent dans la zone d'affichage :



Vous pouvez double-cliquer sur un exemple pour l'insérer directement dans la zone Définition.

### Filtres de saisie :

- *Caractères d'initialisation pour l'affichage* : modification des caractères d'emplacement
- *Caractères d'initialisation* : caractères d'initialisation des filtres
- *Codes de début* : codes de définition des filtres
- *Raccourcis* : raccourcis de filtres
- *Caractères d'emplacement* : caractères utilisés pour désigner les emplacements possibles
- *Les principaux caractères ignorés* : caractères ignorés dans les filtres
- *Exemples de filtres de dates*
- *Exemples de filtres d'heures*



- *Exemples de filtres de numéros de téléphone*
- *Exemple de filtre du numéro de sécurité sociale*
- *Autres exemples de filtres de saisie*

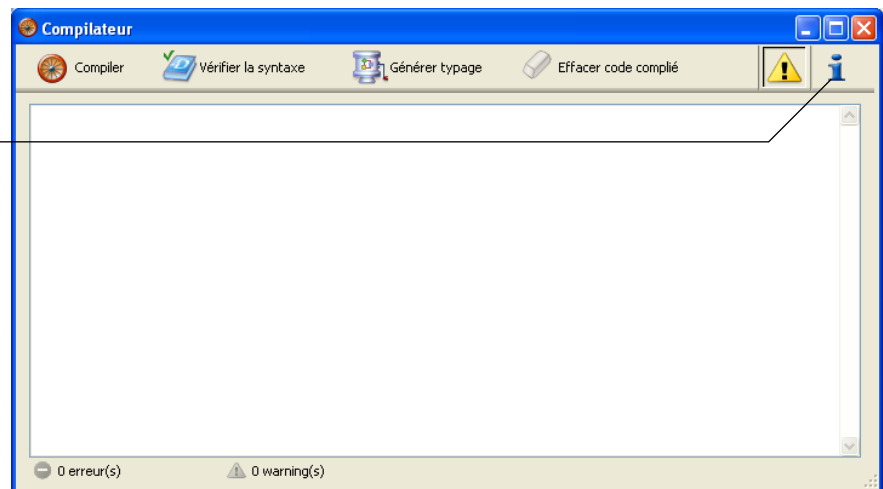
#### **Formats d'affichage :**

- *Caractères d'emplacement* : caractères utilisés pour les emplacements vide
- *Les principaux caractères ignorés* : caractères ignorés dans les formats
- *Exemples de formats de dates*
- *Exemples de formats d'heures*
- *Exemples de formats de numéros de téléphone*
- *Exemple de format du numéro de sécurité sociale*

## **Compilateur**

L'interface de la fenêtre de compilation a été modifiée. La fenêtre propose cependant toutes les fonctions précédentes. Un nouveau bouton permet en outre d'accéder directement aux Préférences :

Affichage de la page  
Compilateur des  
Préférences



## **Désactivation des Warning durant la compilation**

Il est désormais possible de désactiver sélectivement certains Warnings lors de la compilation. Pour cela, il suffit d'insérer dans le code d'une méthode 4D :

```
`%W-<numéro de warning>
```

Les numéros de Warning sont maintenant précisés à la suite du message dans la liste des erreurs de compilation.

Par exemple, pour désactiver le Warning suivant :  
*1: Pointeur dans une déclaration de tableau (518.5)*

... il suffit d'écrire dans une méthode 4D, de préférence une méthode *COMPILER\_XXX* (méthode compilée en premier lieu), le commentaire suivant :

```
`%W-518.5
```

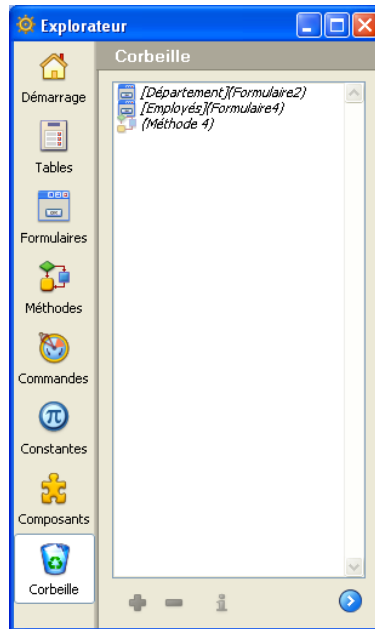
## Explorateur

Outre une interface remaniée, l'Explorateur de 4<sup>e</sup> Dimension 2004 bénéficie de plusieurs nouveautés :

- nouvelle page **Démarrage** permettant d'accéder aux dossiers définis dans la base — cette nouveauté fait l'objet du [paragraphe "Utiliser des dossiers dans l'Explorateur"](#), page 96,
- nouvelle page **Corbeille**, permettant de gérer les objets supprimés,
- affichage détaillé des commandes de plug-ins et des composants dans la page **Composants**,
- nouvelles fonctions de prévisualisation des méthodes et d'affichage de leurs propriétés.

## Page Corbeille

La nouvelle page **Corbeille** de l'Explorateur rassemble tous les objets de type Dossier, Table, Formulaire ou Méthode projet ayant été supprimés dans la base :



A l'image de la corbeille de Windows ou de MacOS, la Corbeille de 4<sup>e</sup> Dimension est un élément de sécurité de l'interface, réduisant le risque de suppression accidentelle d'objets.

Les objets placés dans la Corbeille n'apparaissent plus dans les éditeurs ou les menus de 4<sup>e</sup> Dimension, ils ne peuvent plus être modifiés, utilisés, déplacés, etc. D'autres objets similaires peuvent être créés avec le même nom.

Les objets placés dans la Corbeille peuvent cependant toujours être récupérés tant que la Corbeille n'est pas vidée (cf. ci-dessous).

---

*4D Server* Le contenu de la Corbeille est identique pour tous les postes clients.

---

## Suppression des objets

Les objets peuvent être supprimés depuis la page **Démarrage** ou les pages **Formulaires** et **Méthodes**.

Pour supprimer un objet, il suffit de le sélectionner et de choisir la commande **Supprimer** dans le menu contextuel de l'Explorateur (**clic droit** sous Windows ou **Control+clic sous** MacOS), de le faire glisser

dans la corbeille ou de cliquer sur l'icône de suppression située en bas de la zone :



Dans la page **Démarrage**, il est également possible de choisir la Corbeille comme destination d'un déplacement d'objet.

Il est possible de placer une table dans la Corbeille sauf lorsque la base ne contient qu'une table. La table disparaît alors des éditeurs de la base, y compris de l'éditeur de Structure. Toutefois, il n'est pas possible de supprimer physiquement la table de la base.

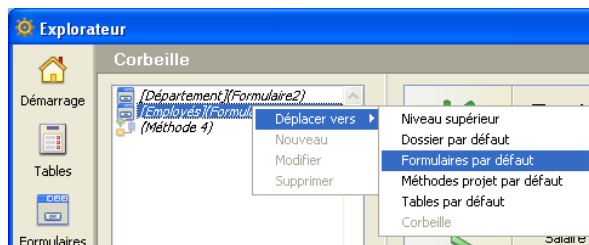
Lorsque vous supprimez une table, les formulaires de la table sont automatiquement supprimés.

Il n'est pas possible de supprimer un formulaire ayant l'attribut E (formulaire entrée courant), S (formulaire sortie courant) ou D (formulaire cumulant les deux attributs) sauf lorsque la table de ce formulaire est supprimée.

Enfin, il n'est pas possible de supprimer un dossier contenant un de ces objets non-supprimables.

### Récupérer un objet

Vous pouvez à tout moment récupérer un ou plusieurs objet(s) placé(s) dans la Corbeille (tant qu'elle n'a pas été vidée). Pour cela, il suffit de sélectionner le ou les objet(s) puis de le(s) faire glisser vers la barre d'icônes située à gauche de la fenêtre ou de choisir la commande **Déplacer vers>** dans le menu contextuel de la liste :

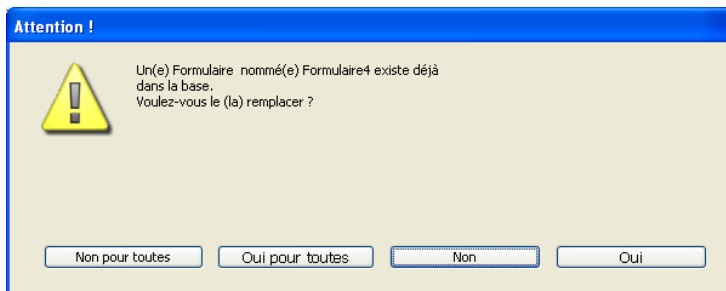


Le sous-menu **Déplacer vers>** liste les dossiers présents dans la base (pour plus d'informations sur les dossiers, reportez-vous au [paragraphe "Utiliser des dossiers dans l'Explorateur"](#), page 96). Si la base ne contient pas de dossiers, seul le libellé **Niveau supérieur** est disponible.

Lorsque vous choisissez une commande de ce sous-menu, la sélection d'objets est récupérée et replacée dans le dossier choisi. Si vous choisissez **Niveau supérieur**, la sélection est placée au premier niveau, en-dehors de tout dossier.

Si un ou plusieurs objets de la base portent le même nom qu'un ou plusieurs objets que vous souhaitez récupérer, 4<sup>e</sup> Dimension affiche une boîte de dialogue d'alerte proposant :

- de remplacer l'objet dans la base (formulaires, méthodes, etc.).
- de renommer *nom\_1* l'objet à récupérer (tables),



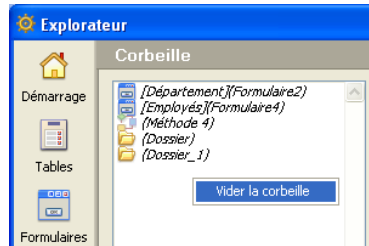
Le bouton **Oui** renomme l'objet en cours de récupération ou remplace l'objet existant. Le bouton **Oui pour toutes** renomme ou remplace sur le même principe tous les objets de la sélection (le cas échéant). Si vous cliquez sur **Non** ou **Non pour toutes**, la récupération d'objets n'est pas effectuée.

### Vider la corbeille

Par défaut, les objets supprimés restent dans la corbeille tant qu'elle n'est pas vidée. Vous pouvez à tout moment décider de vider la corbeille, par exemple avant un compactage, dans le but de réduire la taille du fichier de structure. Tous les objets présents dans la corbeille sont alors définitivement supprimés de la base.

A noter que les tables et les formulaires courants (avec l'attribut E, S ou D) ne peuvent pas être supprimés.

Pour vider la corbeille, choisissez la commande **Vider la corbeille** dans le menu contextuel de l'Explorateur (**clic droit** sous Windows ou **Control+clic sous MacOS**) :



---

*Note* Vous devez cliquer en-dehors de tout objet.

---

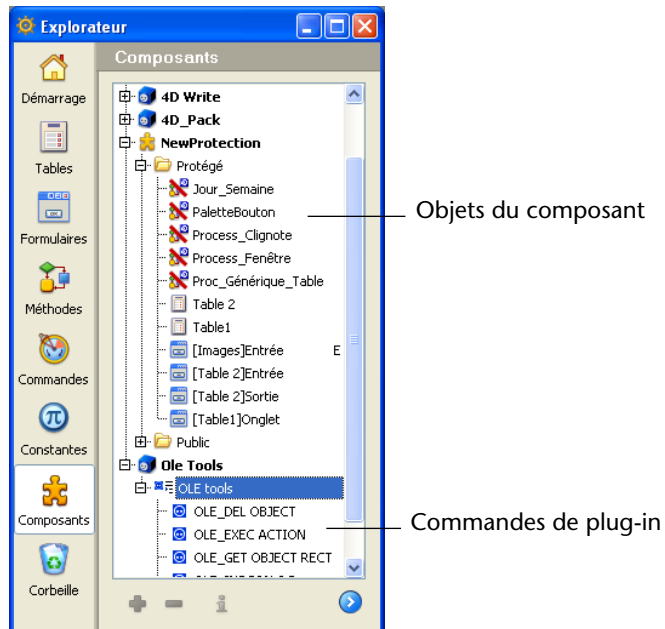
Une boîte de dialogue d'alerte indiquant le nombre d'objets "supprimables" présents dans la corbeille vous permet de confirmer ou d'annuler l'opération :



Cliquez sur **OK** pour supprimer tous les objets de la Corbeille.

## Affichage détaillé dans la page Composants

La page **Composants** de l'Explorateur propose désormais un affichage détaillé des plug-ins et des composants contenus dans la base :

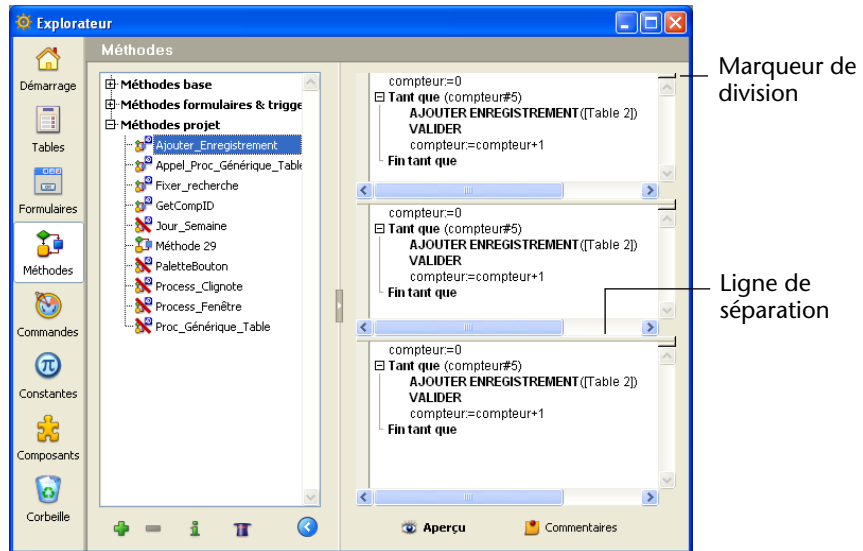


- Les commandes des plug-ins apparaissent désormais, classées par thèmes. Vous pouvez les insérer dans vos méthodes par glisser-déposer.
- Les objets des composants installés (tables, formulaires et méthodes projet) apparaissent sous forme de listes hiérarchiques, classées par composant et par type (Public et Protégé). Il est possible de prévisualiser les formulaires, les tables et les méthodes publiques.

## Prévisualisation des méthodes et affichage des propriétés

Désormais, la zone de prévisualisation des méthodes affiche les couleurs associées aux objets et peut être divisée en plusieurs zones

indépendantes dotées de barres de défilement, à l'image des fenêtres de l'éditeur de méthodes :



Pour ajouter une zone, cliquez sur le marqueur de division. Pour supprimer une partie, double-cliquez sur la ligne de division.

L'Explorateur contient deux nouvelles icônes. La première, activée uniquement lorsqu'une méthode projet est sélectionnée, affiche la fenêtre des Propriétés de la méthode. La seconde affiche l'assistant Web Services (permettant de générer des méthodes projet proxy) :

Affichage des propriétés de la méthode —   — Affichage de l'assistant Web services

## Utiliser des dossiers dans l'Explorateur

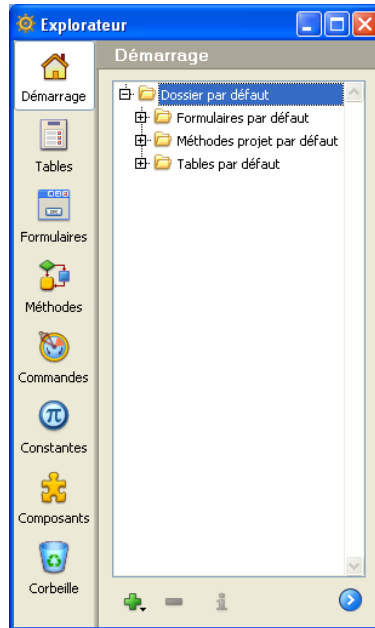
Il est possible de définir et d'utiliser des "dossiers" d'objets dans l'Explorateur de 4<sup>e</sup> Dimension 2004. A l'instar des dossiers utilisables dans les systèmes d'exploitation Windows et MacOS, les dossiers de 4<sup>e</sup> Dimension permettent de regrouper différents objets en fonction de critères personnalisés (notamment par fonctionnalité) et non uniquement par nature. Par exemple, vous pouvez rassembler au sein d'un dossier nommé "Messagerie" les objets relatifs à la prise en charge de la fonction de messagerie dans votre base.



Dans 4<sup>e</sup> Dimension, les objets de type Table, Formulaire et Méthode projet peuvent être placés dans des dossiers. Il est également possible de créer des sous-dossiers sur plusieurs niveaux.

## Présentation

La gestion des dossiers s'effectue via une liste hiérarchique dans la page **Démarrage** de l'Explorateur de 4<sup>e</sup> Dimension :



Tous les objets personnalisés de la base (c'est-à-dire les tables, les formulaires et les méthodes projet) sont toujours affichés dans la page. A noter qu'ils restent cependant listés dans les autres pages de l'Explorateur.

Les dossiers se présentent sous forme d'éléments d'une liste hiérarchique. Chaque dossier peut contenir n'importe quel type d'objet personnalisé ainsi que d'autres dossiers (appelés "sous-dossiers").

Dans cette page, vous pouvez ajouter des dossiers ou des sous-dossiers, renommer, déplacer et réorganiser les dossiers et les objets. Vous pouvez également créer directement des tables, des formulaires et des méthodes projet. Comme dans les autres pages de l'Explorateur, vous pouvez ouvrir une table, un formulaire ou une méthode projet dans une fenêtre de l'éditeur correspondant en double-cliquant sur son

nom, ou l'insérer par glisser-déposer. Enfin, vous pouvez supprimer des dossiers et des objets.

Il est également possible d'assigner un dossier à chaque objet au moment de sa création.

---

*4D Server* Tous les postes clients connectés à une même base partagent la même configuration de dossiers. Dès qu'une modification est effectuée par un poste client dans la page Démarrage (déplacement, ajout, etc.), elle est automatiquement et instantanément répercutée sur tous les postes clients connectés.

---

### Pourquoi utiliser des dossiers ?

Les dossiers sont utiles pour organiser vos bases par fonctionnalités ou suivant des critères personnalisés (dates, langues, etc.). L'accès aux différents objets est alors facilité depuis l'Explorateur.

En outre, les dossiers sont accessibles sous forme de listes hiérarchiques depuis l'éditeur de méthodes de 4<sup>e</sup> Dimension. Ce principe facilite l'appel aux objets adéquats lors de l'écriture des méthodes :



Enfin, les dossiers de 4<sup>e</sup> Dimension sont compatibles avec les groupes de 4D Insider et inversement ; vous pouvez ainsi facilement déplacer des fonctionnalités entières entre deux bases. Pour plus d'informations sur ce point, reportez-vous au [paragraphe "Correspondance avec les groupes de 4D Insider"](#), page 105.

### Compatibilité avec le fonctionnement précédent

Les dossiers constituent un nouveau mode de classement des objets d'une base et ne se substituent pas à la hiérarchie traditionnelle des objets. Leur utilisation n'est pas obligatoire. Pour tirer parti des

dossiers, il suffit d'utiliser la page **Démarrage** de l'Explorateur. Cependant, les pages **Tables**, **Formulaires**, **Méthodes**, etc. restent accessibles et fonctionnent comme dans les versions précédentes de 4<sup>e</sup> Dimension.

## Dossiers par défaut

Dans le cas des bases sans dossiers (bases converties depuis des versions précédentes de 4<sup>e</sup> Dimension ou nouvelles bases dans lesquelles l'utilisateur ne crée pas de dossier), 4<sup>e</sup> Dimension crée un "Dossier par défaut" contenant trois sous-dossiers nommés Formulaires par défaut, Méthodes projet par défaut et Tables par défaut.

- Lors de la création d'une nouvelle base, aucun dossier par défaut n'est créé ; si l'utilisateur ne crée pas de dossier, 4<sup>e</sup> Dimension crée le dossier et les sous-dossiers par défaut au lancement suivant de la base et y place les objets créés par l'utilisateur. Si l'utilisateur crée au moins un dossier, 4<sup>e</sup> Dimension ne crée pas de dossier par défaut.
- Lors de l'ouverture d'une base créée avec une version précédente de 4<sup>e</sup> Dimension et convertie en version 2004, le programme crée les dossiers par défaut et y place tous les objets de la base, sauf si des groupes 4D Insider avaient été définis — auquel cas 4<sup>e</sup> Dimension ne fait rien.

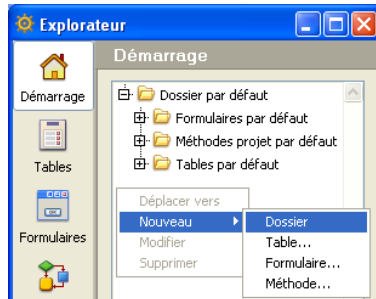
Ce dossier par défaut a uniquement pour but de ranger les objets existants, les sous-dossiers qu'il contient ne possèdent pas de propriétés particulières. Vous pouvez les modifier, les supprimer ou les renommer comme vous le souhaitez.

Par la suite, tous les nouveaux objets créés dans la base seront par défaut placés au premier niveau de la liste des dossiers, c'est-à-dire en-dehors de tout dossier ("Niveau supérieur"). Il vous appartient de créer les dossiers nécessaires et de gérer leur contenu.

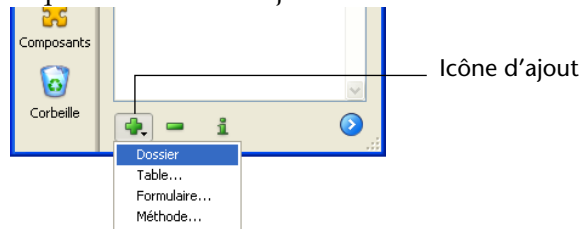
## Créer des dossiers ou des sous-dossiers

Vous pouvez à tout moment ajouter des dossiers ou des sous-dossiers dans la page **Démarrage** de l'Explorateur.

Pour cela, il suffit d'utiliser la commande **Nouveau>Dossier** du menu contextuel de la zone (**clik droit** sous Windows, **Control+clik** sous MacOS) :



Vous pouvez également utiliser le menu qui apparaît lorsque vous cliquez sur l'icône d'ajout située en-dessous de la zone :



Le dossier ou le sous-dossier est immédiatement créé et son nom est par défaut "*Dossier\_ +numéro du dossier*". Vous pouvez modifier ce nom comme vous le souhaitez — le nom d'un dossier ou d'un sous-dossier doit simplement être unique.

- Pour créer un dossier, cliquez en-dehors de tout dossier existant dans la zone ou assurez-vous qu'aucun élément n'est sélectionné au moment où vous cliquez sur l'icône d'ajout.
- Pour créer un sous-dossier, cliquez sur le dossier au sein duquel vous souhaitez ajouter le sous-dossier ou sélectionnez-le avant de cliquer sur l'icône d'ajout.

Par la suite, vous pourrez facilement transformer un dossier en sous-dossier et inversement à l'aide du glisser-déposer ou des commandes de déplacement (cf. ci-dessous).

### Renommer un dossier ou un objet

Vous pouvez à tout moment renommer un dossier (ou tout objet de la page **Démarrage**). Pour cela, effectuez **Alt+clik** (Windows) ou **Option+clik** (MacOS) sur son nom. Vous pouvez également cliquer deux fois sur l'élément (attendez quelques instants entre les deux clics afin de ne pas provoquer de double-clic).

L'élément passe en édition :

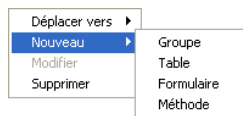


Effectuez vos modifications puis cliquez en-dehors de la zone ou appuyez sur la touche **Tab** pour valider la saisie.

- Notes*
- Le nom d'un dossier est limité à 31 caractères.
  - Deux dossiers ne peuvent pas avoir le même nom (quel que soit leur niveau hiérarchique).
  - Il n'est pas possible de renommer une table en modifiant son nom placé devant un nom de formulaire (par exemple [Matable]Form).

## Créer ou modifier un objet

La commande **Nouveau**> du menu contextuel permet de créer des dossiers (cf. [paragraphe "Créer des dossiers ou des sous-dossiers", page 99](#)), mais également des tables, des formulaires et des méthodes projets.



Lorsque vous choisissez une sous-commande de ce type, vous accédez directement à la boîte de dialogue de création de l'objet (cf. [paragraphe "Ajouter un objet à un dossier lors de sa création", page 103](#)).

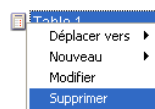
La commande **Modifier...** du menu contextuel vous permet d'afficher les objets sélectionnés dans une fenêtre de l'éditeur correspondant (tables, formulaires et méthodes projet).



Cette commande est grisée lorsque l'élément sélectionné est un dossier ou un sous-dossier, ou lorsque plusieurs éléments sont sélectionnés.

## Supprimer un objet

La commande **Supprimer** provoque la suppression de la sélection. Si la sélection comporte un dossier ou un sous-dossier, tous les éléments qu'il contient sont supprimés :



Cette commande peut être utilisée avec tous les types d'éléments (dossiers, sous-dossiers, tables, formulaires et méthodes projet) et aux sélections multiples. Les objets supprimés disparaissent des listes et des éditeurs de l'application.

Les éléments supprimés sont en réalité placés dans la **Corbeille** de l'Explorateur. Pour plus d'informations sur le fonctionnement de la corbeille de l'Explorateur, reportez-vous au [paragraphe "Page Corbeille", page 91](#).

Pour supprimer un ou plusieurs objet(s) depuis la page Départ de l'Explorateur, il est également possible de le(s) sélectionner puis de choisir la commande **Déplacer vers>Corbeille** du menu contextuel de la zone (**clic droit** sous Windows, **Control+clic** sous MacOS).

### Organiser les dossiers et les sous-dossiers

Vous pouvez à tout moment réorganiser le contenu des dossiers et des sous-dossiers. Vous pouvez placer des objets dans des dossiers, des dossiers dans d'autres dossiers, etc. Si vous placez un dossier dans un autre dossier, il devient un sous-dossier. Les caractéristiques des dossiers et des sous-dossiers sont identiques.

Les dossiers sont déplacés avec leur contenu. La hiérarchie des objets déplacés est conservée dans le dossier d'arrivée.

Vous pouvez déplacer des objets seuls ou des sélections d'objets. Pour créer une sélection continue d'objets, appuyez sur la touche **Maj** et cliquez sur le premier et le dernier objet à sélectionner. Pour effectuer une sélection discontinue d'objets, cliquez sur chaque objet à sélectionner en maintenant la touche **Ctrl** (Windows) ou **Commande** (MacOS) enfoncée.

Pour organiser le contenu des dossiers, vous disposez de trois possibilités :

- utiliser le glisser-déposer,
- utiliser la commande de déplacement du menu contextuel,
- utiliser le menu **Dossier** au moment de la création de chaque objet.

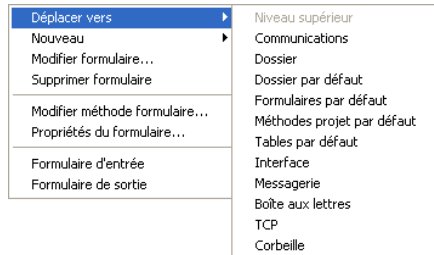
### Glisser-déposer

Vous pouvez organiser le contenu des dossiers dans la page **Démarrage** de l'Explorateur à l'aide du glisser-déposer. Pour déplacer un dossier, un objet ou une sélection, il suffit de cliquer dessus et, tout en maintenant le bouton de la souris enfoncé, le faire glisser et le déposer à un autre emplacement.

## Déplacer vers

La commande **Déplacer vers** du menu contextuel de la page **Démarrage** permet de modifier le contenu d'un dossier sans devoir au préalable déplacer le dossier de destination.

Pour cela, cliquez avec le **bouton droit** de la souris (Windows) ou effectuez **Control+clic** (MacOS) sur l'objet, le dossier ou la sélection. Le menu contextuel s'affiche. La commande hiérarchique **Déplacer vers** propose comme destinations possibles la totalité des dossiers et sous-dossiers existants :



Il vous suffit de sélectionner le dossier dans lequel la sélection sera déplacée. Si vous choisissez **Niveau supérieur**, la sélection sera placée au premier niveau de la liste, en-dehors de tout dossier (cette ligne est grisée lorsque vous déplacez déjà un dossier de premier niveau).

Si vous choisissez **Corbeille**, la sélection sera supprimée (cf. [paragraphe "Supprimer un objet", page 101](#)).

## Ajouter un objet à un dossier lors de sa création

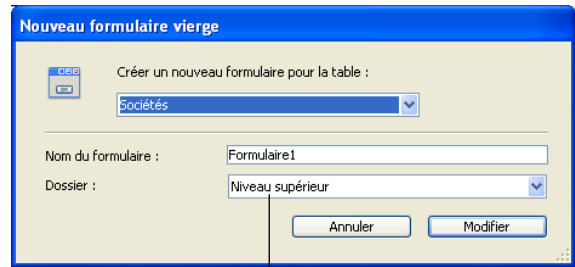
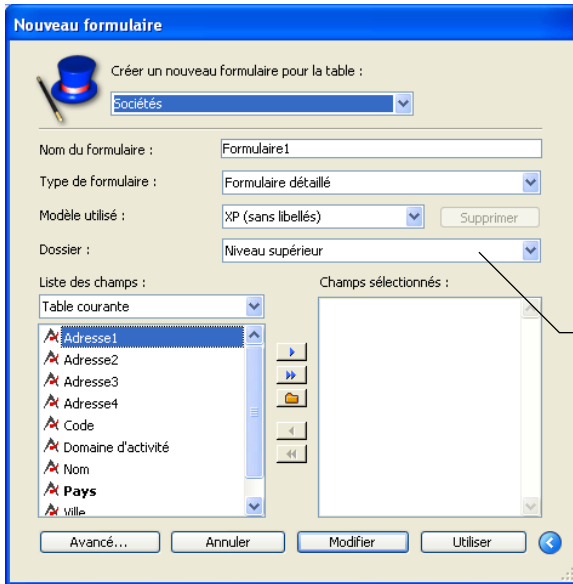
Vous pouvez placer directement un nouvel objet dans un dossier au moment de sa création. Un nouveau menu, situé dans les boîtes de dialogue de création de tables, de formulaires et de méthodes projet, permet de désigner le dossier dans lequel l'objet doit être créé. Par défaut, les objets sont créés au premier niveau (Niveau supérieur) ou dans le dernier dossier sélectionné dans l'Explorateur (le cas échéant).

### ■ Boîte de dialogue de création de table :



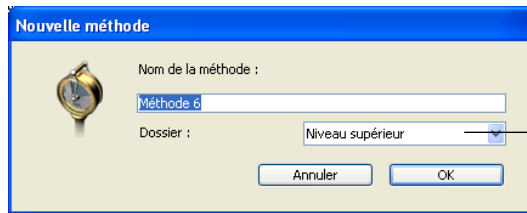
Menu de sélection du groupe dans lequel ajouter la table

■ Boîtes de dialogue de création de formulaires :



Menu de sélection du dossier dans lequel ajouter le formulaire

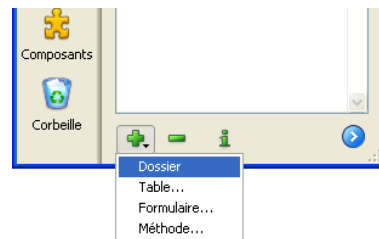
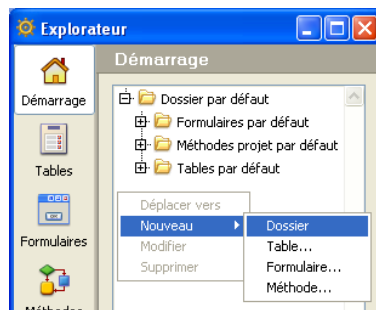
■ Boîte de dialogue de création de méthode projet :



Menu de sélection du dossier dans lequel ajouter la méthode

Bien entendu, vous pourrez toujours changer ces objets de dossier après leur création dans la page **Démarrage** de l'Explorateur.

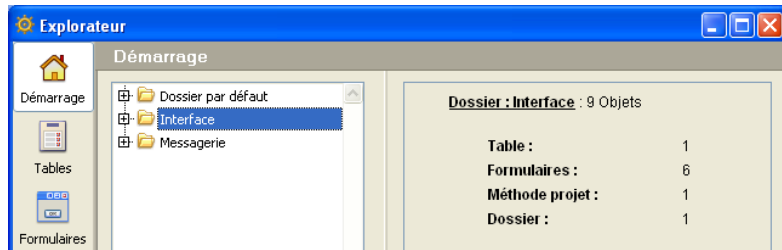
A noter que vous pouvez demander la création de nouveaux objets (tables, formulaires et méthodes projet) directement depuis la page **Démarrage** de l'Explorateur via le menu contextuel ou l'icône d'ajout :





## Informations sur le dossier

Lorsqu'un dossier est sélectionné dans l'Explorateur, la zone de prévisualisation de la fenêtre affiche plusieurs informations :



- **Dossier : Nom du dossier** : Nombre d'objets disponibles à l'intérieur du dossier
- **Table(s)** : Nombre de tables contenues dans le dossier
- **Formulaire(s)** : Nombre de formulaires contenus dans le dossier
- **Méthode(s) projet** : Nombre de méthodes projet contenues dans le dossier
- **Dossier(s)** : Nombre de dossiers (sous-dossiers) contenus dans le dossier

Si un type d'objet n'est pas présent dans un dossier, il n'apparaît pas dans la zone de prévisualisation. Si un dossier est vide, la zone affiche uniquement **Dossier : Nom du dossier**

Lorsque plusieurs dossiers et/ou objets sont sélectionnés, les informations affichées concernent l'élément courant, c'est-à-dire le dernier sur lequel vous avez cliqué.

## Correspondance avec les groupes de 4D Insider

Tous les dossiers créés dans 4<sup>e</sup> Dimension sont visibles par 4D Insider sous forme de groupes et inversement. Les mêmes opérations peuvent leur être appliquées. En fait, ces deux types de structures sont semblables.

Toutefois, les groupes créés par 4D Insider peuvent contenir d'autres types d'objets : barres de menus et menus, énumérations, feuilles de style, formats et filtres, images, messages d'aide (non liés à des champs), méthodes base et ressources STR#.

Ces objets ne sont pas visibles dans l'Explorateur de 4<sup>e</sup> Dimension. Par conséquent, lors de la suppression d'un dossier/groupe dans 4<sup>e</sup> Dimension, les objets non visibles qu'il contient éventuellement ne sont pas supprimés. Ils sont automatiquement extraits du dossier et

restent accessibles dans les différents éditeurs. Côté 4D Insider, ces objets sont placés au premier niveau de la base (hors groupes).

En revanche, certains objets sont indissociables de leur objet "hôte" et, bien que visibles uniquement par 4D Insider, ils sont supprimés en même temps que l'objet lié :

- les méthodes formulaire sont supprimées avec leurs formulaires
- les méthodes objet sont supprimées avec leurs formulaires
- les messages d'aide liés à des champs sont supprimés avec les tables
- les triggers sont supprimés avec les tables.

## Editeur de méthodes

Trois principales nouveautés sont disponibles dans l'éditeur de méthodes de 4<sup>e</sup> Dimension 2004 :

- la macro-commande **method**, permettant d'appeler une méthode,
- la prise en charge du dossier Macros,
- la nouvelle barre d'outils des fenêtres de l'éditeur.

### Macro d'appel de code 4D

Il est désormais possible dans l'éditeur de méthodes de 4<sup>e</sup> Dimension 2004 de générer et d'utiliser des macro-commandes exécutant des méthodes projet 4D. Ce principe autorise notamment les développeurs à créer des fonctions sophistiquées diffusées via des macro-commandes associées à des composants.

### Balise <method>

Pour appeler du code 4D au sein d'une méthode, une nouvelle balise double a été ajoutée dans le langage des macro-commandes de 4<sup>e</sup> Dimension 2004 : **<method>** **</method>**

Ces balises indiquent le début et la fin du nom d'une méthode projet et de son paramètre (facultatif). La méthode est exécutée au moment de l'appel de la macro. Vous pouvez lui passer un paramètre sous la forme ("param1;param2;..."). Ce paramètre sera reçu dans la méthode via les variables \$1, \$2, etc. Par exemple, la macro suivante provoquera l'exécution de la méthode *MaMéthode* avec le nom de la méthode courante comme paramètre :

```
<method>MaMéthode("<method_name/>")</method>
```

**Contexte d'exécution de la méthode appelée** Le code de la méthode appelée est exécuté dans un nouveau process. Ce process est tué à l'issue de l'exécution de la méthode.

---

*Le process de structure reste figé tant que l'exécution de la méthode appelée n'est pas terminée. Vous devez donc veiller à ce que celle-ci soit rapide et ne risque pas de provoquer de blocage de l'application. Si ce cas se produit, utilisez la combinaison Ctrl+F8 (Windows) ou Commande+F8 (MacOS) permettant de "tuer" un process.*

---

Il est possible d'utiliser des variables process spéciales, gérées automatiquement par 4<sup>e</sup> Dimension, dans la méthode appelée. Certaines variables sont remplies par 4<sup>e</sup> Dimension (variables entrée), la valeur des autres doit être assignée par le développeur (variables sortie) :

Variables entrée	Type		Description
_textSel	C_TEXTE	<--	Texte sélectionné (vide si > 32000 cars.)
_blobSel	C_BLOB	<--	Texte sélectionné
_selLen	C_ENTIER LONG	<--	Longueur du texte sélectionné
_textMethod	C_TEXTE	<--	Texte de la méthode (vide si > 32000 cars.)
_blobMethod	C_BLOB	<--	Texte de la méthode
_methodLen	C_ENTIER LONG	<--	Longueur du texte de la méthode

Variables sortie	Type		Description
_textReplace	C_TEXTE	-->	Résultat à substituer, "" par défaut
_blobReplace	C_BLOB	-->	Résultat à substituer, "" par défaut
_action	C_ENTIER LONG	-->	Type de substitution dans la méthode d'appel : 0 = Aucun 1 = Insérer _textReplace 2 = Insérer _blobReplace 3 = Remplacer la méthode par _textReplace 4 = Remplacer la méthode par _blobReplace

### Exemple

Cet exemple de macro-commande insère automatiquement le code permettant de créer un enregistrement dans la table sélectionnée et d'initialiser chacun de ses champs, quel que soit son type. La macro appelle la méthode *Init\_enregistrement\_macro*, qui doit être accessible dans la base.

- Voici le code de la macro-commande :  

```
<macro name = "Init_enregistrement">
<text>
<method>Init_enregistrement_macro</method>
</text>
</macro>
```
- Voici le code de la méthode projet *Init\_enregistrement\_macro* :

```
C_TEXTE(_textSel;_textReplace)
C_TEXTE($texte)
$texte:=_textSel

$debut:=Position("[";$texte)
Si ($debut>0)
  $texte:=Sous chaine($texte;$debut)
  $fin:=Position("]";$texte)
  Si ($fin>0)
    $texte:=Sous chaine($texte;1;$fin)
    $table:=0
    Boucle ($i;1;Nombre de tables)
      Si ($texte="[ "+Nom de la table($i)+" ]")
        $table:=$i
        $i:=Nombre de tables+1
      Fin de si
    Fin de boucle
    $br:=Caractere(Retour chariot)
    $to_insert:=Nom commande(68)+"("$texte+")"+$br
    `68 = commande CREER ENREGISTREMENT

    Boucle ($i;1;Nombre de champs($table);1)
      $champ:=$texte+Nom du champ($table;$i)
      $type:=Type(Champ($table;$i)->)
      Au cas ou
      : ($type=Est un champ alpha ) | ($type=Est un texte )
        $to_insert:=$to_insert+$champ+":="\\"+$br
      : ($type=Est un numérique ) | ($type=Est un entier ) |
        ($type=Est un entier long )
        $to_insert:=$to_insert+$champ+":="0"+$br
      : ($type=Est une date )
        $to_insert:=$to_insert+$champ+":="!00/00/00!"+$br
      : ($type=Est une heure )
        $to_insert:=$to_insert+$champ+":="?00:00:00?"+$br
      : ($type=Est un booléen )
        $to_insert:=$to_insert+$champ+":=""+Nom commande(215)+$br
        `215 = commande Faux
```

```

: ($type=Est une image)
  $to_insert:=$to_insert+Nom commande(286)+"($pict)+"$br
  `286 = commande C_IMAGE
  $to_insert:=$to_insert+$champ+":"=$pict"+"$br
: ($type=Est un BLOB)
  $to_insert:=$to_insert+Nom commande(604)+"($blob)+"$br
  `604 = commande C_BLOB
  $to_insert:=$to_insert+$champ+":"=$blob"+"$br
Fin de cas
Fin de boucle

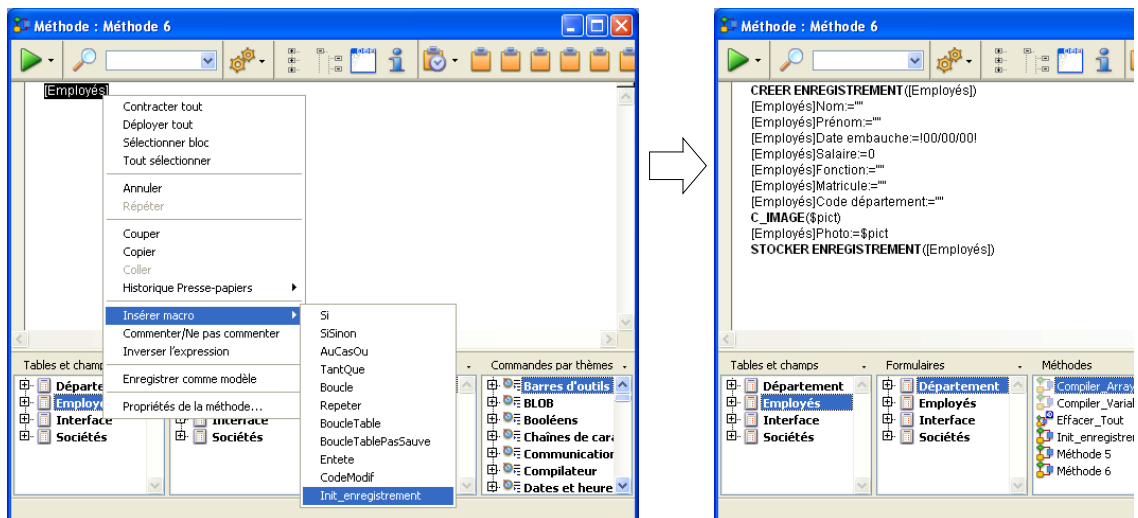
```

```

_textReplace:=$to_insert+Nom commande(53)+"("$texte+")"$br
  `53 = commande STOCKER ENREGISTREMENT
_action:=1
Sinon
  _action:=0
Fin de si
Sinon
  _action:=0
Fin de si

```

- Voici une illustration du fonctionnement de cette macro :



## Nouveau dossier Macros

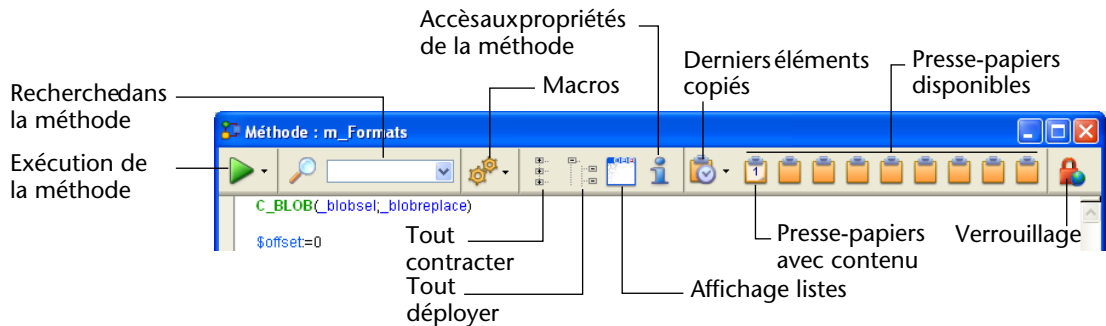
4<sup>e</sup> Dimension 2004 accepte désormais plusieurs documents de macros au format XML. Ces documents doivent être placés dans un dossier nommé **Macros** et placé à côté du fichier "Macros.xml" (situé dans le dossier **4D** actif de la machine). Le nom des documents macros XML supplémentaires est libre.

Le fonctionnement précédent du fichier "Macros.xml" est conservé. 4<sup>e</sup> Dimension charge en premier lieu les macro-commandes du fichier "Macros.xml", puis celles situées dans le dossier Macros s'il y en a un.

Ce mécanisme permet d'utiliser simultanément plusieurs fichiers de macros différents dans une application 4<sup>e</sup> Dimension. Il est ainsi plus facile de diffuser des macro-commandes personnalisées.

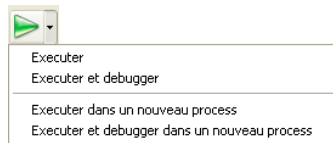
### Nouvelle barre d'outils

Chaque fenêtre de l'éditeur de méthodes dispose d'une barre d'outils permettant d'accéder instantanément aux principales fonctions relatives à l'exécution et l'édition des méthodes :



Les fonctions de la barre d'outils sont les suivantes :

- **Exécution de la méthode** : ce bouton provoque l'exécution de la méthode en mode Utilisation. A l'aide du menu associé au bouton, vous pouvez sélectionner le type d'exécution :



- **Recherche** : l'icône de recherche provoque l'affichage de la boîte de dialogue standard de recherche, permettant d'effectuer une recherche dans la méthode courante.
- **Macros** : ce bouton affiche un menu listant toutes les macros-commandes disponibles.
- **Derniers éléments** : ce bouton affiche un menu listant les 20 derniers éléments copiés dans la fenêtre. Si vous sélectionnez un élément, il est recopié à l'emplacement du curseur.

- **Presse-papiers** : ces 9 icônes représentent les 9 presse-papiers disponibles dans l'éditeur de méthodes. Une icône blanche contenant un chiffre indique un presse-papiers comportant des données, une icône orange indique un presse-papiers vide.
  - Pour copier du texte dans un presse-papiers, sélectionnez-le dans la méthode puis effectuez **Maj+clic** sur l'icône du presse-papiers à utiliser ou utilisez le raccourci **Ctrl+Maj+numéro du presse-papiers** (Windows) ou **Commande+Maj+numéro du presse-papiers** (MacOS). A noter que vous devez utiliser les touches numérotées du clavier alphanumérique.
  - Pour coller le contenu d'un presse-papiers à l'emplacement du curseur, cliquez sur son icône ou utilisez le raccourci **Ctrl+numéro du presse-papiers** (Windows) ou **Commande+numéro du presse-papiers** (MacOS). A noter que vous devez utiliser les touches numérotées du clavier alphanumérique.
- **Tout déployer / Tout contracter** : ces boutons permettent de déployer et de contracter toutes les structures de programmation de la méthode.
- **Affichage listes** : ce bouton permet alternativement d'afficher ou de masquer les listes dans la fenêtre.
- **Verrouillage** : ce bouton indique que la méthode est verrouillée par un autre utilisateur en client/serveur. Cliquer sur le cadenas permet de recharger la méthode (et donc d'afficher les éventuelles modifications effectuées par l'autre utilisateur).
- **Accès aux propriétés** : ce bouton provoque l'affichage de la boîte de dialogue des Propriétés de la méthode.

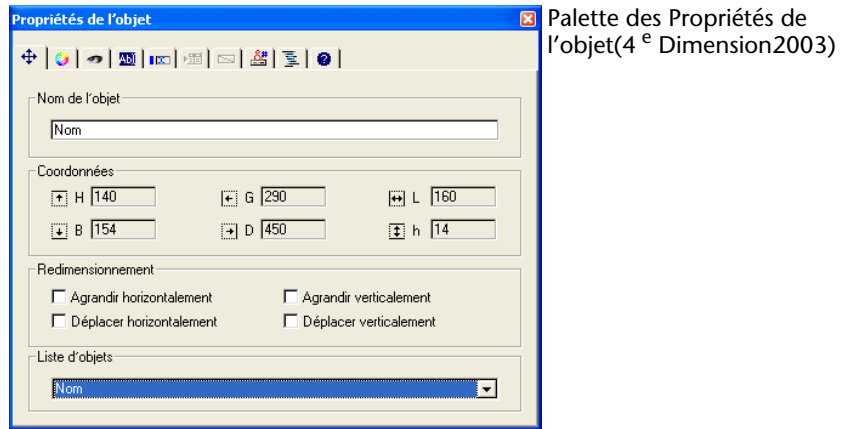
## Editeur de formulaires

Outre l'interface générale, de nombreux aspects de l'éditeur de formulaires de 4<sup>e</sup> Dimension 2004 ont été modifiés :

- suppression de la palette "Propriétés de l'objet" et modification de la palette "Liste des propriétés",
- nouvelles barres d'objets et d'outils intégrées dans chaque fenêtre,
- modification de la définition des paramètres d'impression des formulaires,
- nouveaux outils : zoom, vues, badges, modèles d'objets et nouvelle fonction de magnétisme.

## Palette et Liste des propriétés

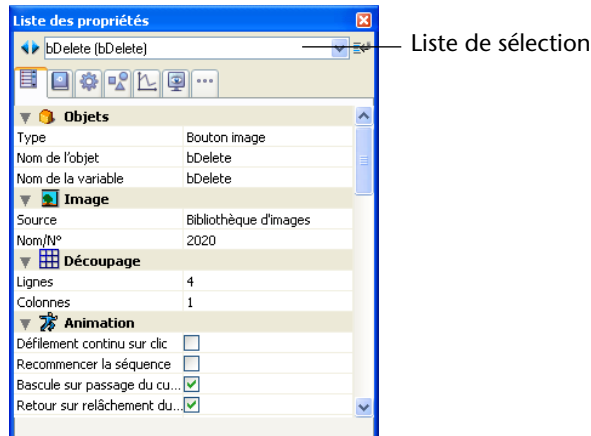
La palette des Propriétés de l'objet est supprimée de la version 2004 de 4<sup>e</sup> Dimension. Toutes les spécificités de cette palette étant accessibles via la Liste des propriétés, cette alternative est désormais inutile.



Par ailleurs, la Liste des propriétés propose de nouvelles possibilités dans 4<sup>e</sup> Dimension 2004 : vous pouvez désormais choisir le contenu de la liste de sélection et la trier, vous pouvez également choisir entre l'affichage contextuel ou thématique des propriétés.

## Configurer la liste de sélection

La liste de sélection permet de sélectionner l'objet dont vous souhaitez visualiser les propriétés :

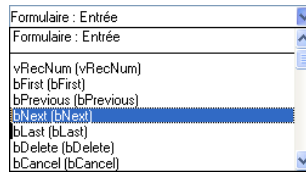


Dans 4<sup>e</sup> Dimension 2004, cette liste contient trois parties, séparées par des filets :

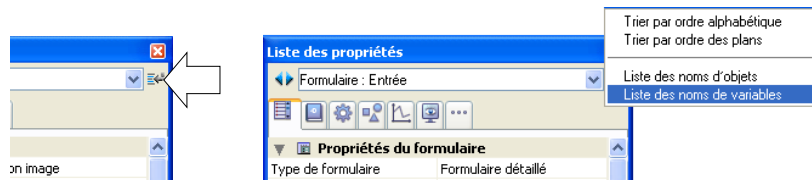
- le nom du formulaire,
- les objets dynamiques (associés à une variable),



- les objets statiques (non associés à une variable).



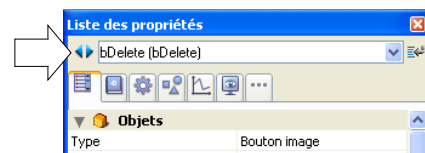
Pour chaque objet dynamique, la liste affiche le nom de la variable suivi, entre parenthèses, du nom d'objet (ou inversement). Vous pouvez définir cet ordre à l'aide du menu associé à la liste :



La commande indique le type de nom affiché en premier lieu.

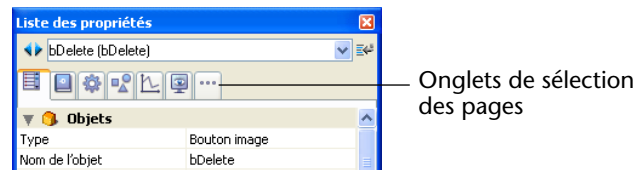
Vous pouvez également modifier le tri de chaque partie de la liste. Par défaut, la liste est triée par ordre alphabétique. Le menu associé vous permet de trier la liste en fonction du plan des objets (du premier plan vers le dernier plan).


Les boutons situés à gauche de la liste vous permettent de sélectionner successivement chaque objet de la liste, vers le haut ou le bas :



## Pages d'affichage des propriétés

Vous pouvez désormais choisir le mode d'affichage des informations dans la Liste des propriétés à l'aide des onglets de sélection des pages :









- La première page  (Général) reprend le fonctionnement de la Liste des propriétés de la version précédente de 4<sup>e</sup> Dimension : elle affiche toutes les informations relatives à l'objet sélectionné via différents thèmes (Source de données, Saisie, Evénements, etc.).

L'affichage est contextuel : seuls les thèmes et les propriétés pertinents pour le type d'objet sélectionné sont affichés. L'affichage de certaines propriétés est également dynamiquement modifié en fonction de la valeur d'autres propriétés.

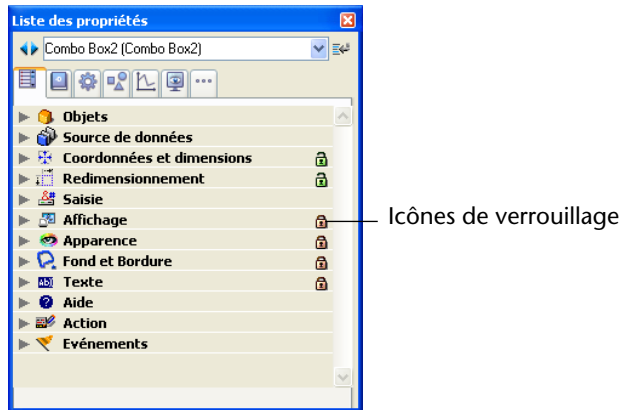
- Les six autres pages contiennent chacune une partie spécifique des informations relatives à l'objet sélectionné. En outre, toutes les propriétés de chaque thème sont affichées, quel que soit le type d'objet. Les propriétés non pertinentes sont grisées.

Cette alternative propose une vision plus globale des possibilités de paramétrage associées aux objets. Voici une description des six pages thématiques :

-  Définition : regroupe les propriétés définissant l'identité de l'objet (type, nom, source de données, etc.).
-  Comportement : regroupe les propriétés définissant le comportement dynamique de l'objet (méthode associée, glisser-déposer, événements formulaires).
-  Dimensions : regroupe les propriétés définissant les coordonnées, la taille et le redimensionnement de l'objet.
-  Valeur : regroupe les propriétés définissant l'édition et l'affichage des objets saisissables (saisissable, équivalent clavier, format d'affichage, etc.).
-  Aspect : regroupe les propriétés définissant l'apparence graphique des objets.
-  Spécifique : regroupe les propriétés spécifiques du type d'objet sélectionné (message d'aide pour les champs, animation pour les boutons image, paramètres d'impression pour les formulaires, etc.). A la différence des autres pages thématiques, les propriétés non pertinentes pour l'objet sélectionné n'apparaissent pas.

## Icônes de verrouillage

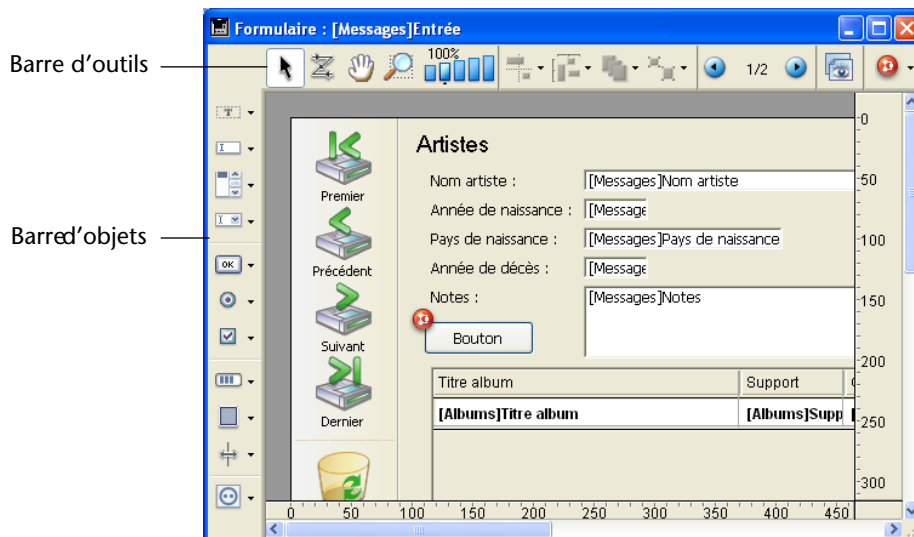
Lorsque la propriété **Modifiable par l'utilisateur** a été appliquée au formulaire, des verrous sont affichés dans la liste des propriétés afin d'indiquer les propriétés modifiables ou non par l'utilisateur :



Pour plus d'informations, reportez-vous au [paragraphe "Formulaires utilisateurs"](#), page 254.

## Barres d'objets et d'outils intégrées

Dans 4<sup>e</sup> Dimension 2004, la palette d'objets et d'outils est désormais intégrée dans chaque fenêtre de l'éditeur de formulaires sous forme de deux barres :



Ce nouveau principe, plus ergonomique, permet un accès direct aux objets et aux principales fonctions.

### Barre d'outils

La nouvelle barre d'outils comporte les éléments suivants :



**Verrouillage utilisateur** : lorsqu'elle est présente à gauche de la barre d'outils, cette icône indique que le formulaire est "Modifiable par l'utilisateur". Dans ce cas il n'est pas possible de le modifier directement, il est nécessaire au préalable de cliquer sur cette icône afin de déverrouiller le formulaire. Pour plus d'informations, reportez-vous au [paragraphe "Modifiable par l'utilisateur", page 130](#).



**Flèche de sélection** (fonctionnement inchangé) : permet de sélectionner, déplacer et redimensionner les objets du formulaire.

*Note* Lorsqu'un objet de type Texte ou Zone de groupe est sélectionné, appuyer sur la touche **Entrée** permet de le passer en édition.

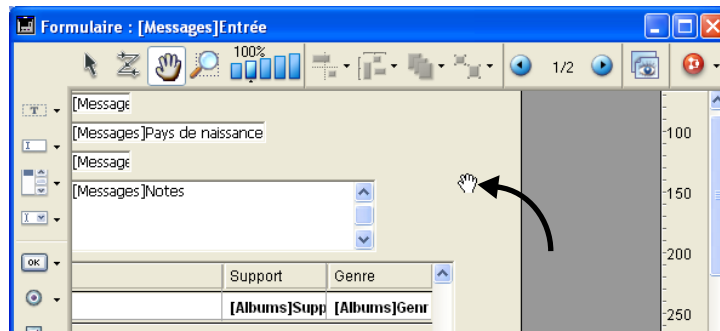


**Ordre de saisie** (fonctionnement inchangé) : passe en mode "Ordre de saisie", dans lequel il est possible de visualiser et de modifier l'ordre de saisie courant du formulaire.

A noter que les *badges* permettent désormais de visualiser l'ordre de saisie courant, tout en travaillant dans le formulaire.




**Déplacement** : passe en mode "Déplacement", dans lequel il est possible d'atteindre rapidement n'importe quelle partie du formulaire en le faisant directement glisser dans la fenêtre. Le curseur prend la forme d'une main :




Ce mode de navigation est particulièrement utile en cas de zoom dans le formulaire.

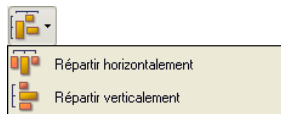



**Zoom** : permet de modifier l'échelle d'affichage du formulaire (100% par défaut). Vous pouvez passer en mode "Zoom" en cliquant sur le bouton loupe ou en cliquant directement sur la barre correspondant à l'échelle désirée. Cette nouvelle fonction est détaillée dans le [paragraphe "Zoom", page 120](#).

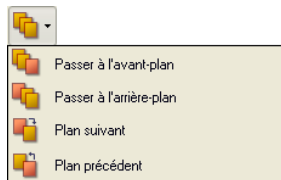
 **Alignement** (fonctionnement inchangé) : ce bouton est associé à un menu permettant d'aligner les objets dans le formulaire.




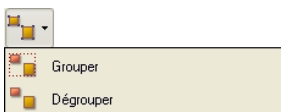
 **Répartition** (fonctionnement inchangé) : ce bouton est associé à un menu permettant de répartir les objets dans le formulaire.



 **Changement de plan** (fonctionnement inchangé) : ce bouton est associé à un menu permettant de modifier le plan des objets dans le formulaire.



 **Groupement/Dégroupement** (fonctionnement inchangé) : ce bouton est associé à un menu permettant de grouper et dégroupier la sélection d'objets du formulaire.



*Note* Il est désormais possible de sélectionner un objet appartenant à un groupe sans devoir dégroupier l'ensemble : pour cela, effectuez **Ctrl+clic** (Windows) ou **Commande+clic** (MacOS) sur l'objet (le groupe doit être sélectionné au préalable).



**Affichage et gestion des pages** : cette zone permet de passer d'une page du formulaire à une autre et d'ajouter des pages. Pour naviguer parmi les pages du formulaire, cliquez sur les boutons fléchés ou

cliquez sur la zone centrale et choisissez la page à afficher dans le menu qui apparaît :



Si vous cliquez sur le bouton fléché de droite alors que vous êtes sur la dernière page du formulaire, 4<sup>e</sup> Dimension vous permet d'ajouter une page.



**Gestion des vues** : ce bouton affiche ou masque alternativement la palette des vues. Cette nouvelle fonction est détaillée dans le [paragraphe “Gestion des vues”, page 121.](#)



**Affichage des badges** : chaque clic sur ce bouton provoque l'affichage successif de tous les types de badges de formulaire. Le bouton est également associé à un menu permettant de sélectionner directement le type de badge à afficher :



Cette nouvelle fonction est détaillée dans le [paragraphe “Utilisation des badges”, page 125.](#)

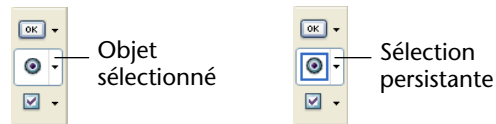
### Barre d'objets

La nouvelle barre d'objets contient tous les objets utilisables dans les formulaires de 4<sup>e</sup> Dimension.

Pour dessiner un type d'objet, sélectionnez le bouton correspondant puis tracez l'objet dans le formulaire.

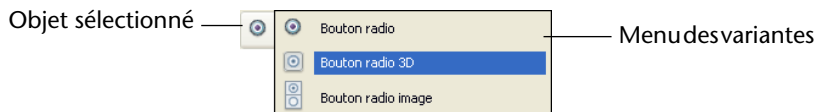
Vous pouvez également cliquer deux fois sur le bouton afin qu'il reste sélectionné même après que vous ayez tracé un objet dans le formulaire.

Cette fonction facilite la création de plusieurs objets de même type à la suite :

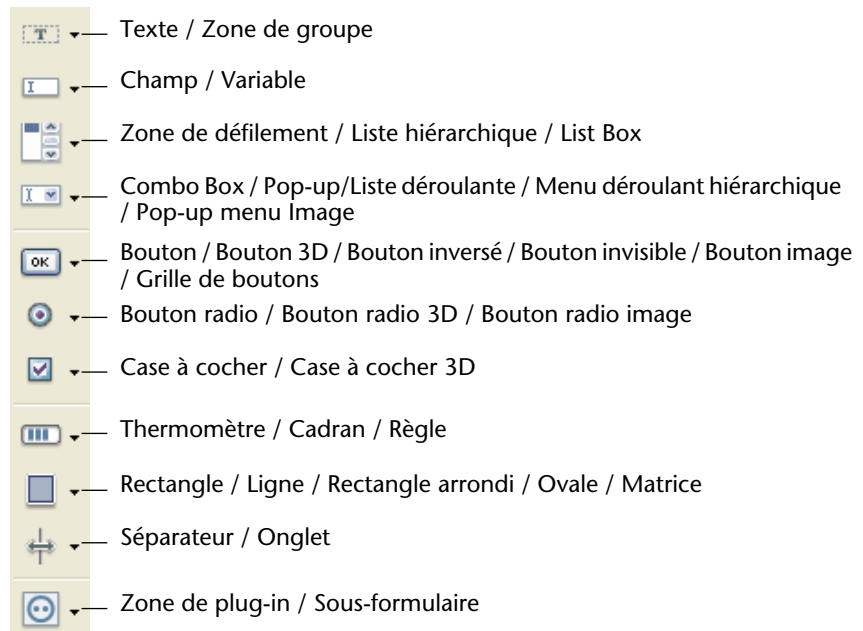


*Note* Pour annuler une sélection persistante, cliquez sur un autre outil.

Enfin, lorsque vous cliquez sur la partie droite du bouton, vous accédez au menu de choix de variante :



La classification thématique des objets a été modifiée par rapport à la précédente version de 4<sup>e</sup> Dimension et de nouveaux objets sont disponibles. Voici les familles d'objets disponibles dans 4<sup>e</sup> Dimension 2004 :



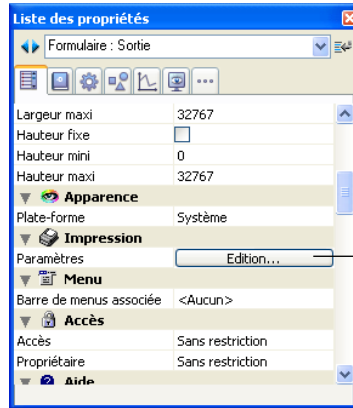
*Note* Les nouveautés relatives aux objets de formulaires sont décrites dans le [paragraphe "Objets de formulaires", page 133](#).

## Paramètres d'impression des formulaires

Le mécanisme de définition et de stockage des paramètres d'impression associés aux formulaires a été modifié dans 4<sup>e</sup> Dimension 2004.

Dans les versions précédentes, ces paramètres étaient définis dans la boîte de dialogue standard de **Format d'impression** accessible depuis le menu **Fichier**. Le formulaire était donc enregistré avec les paramètres d'impression courants utilisés par l'application 4<sup>e</sup> Dimension globale.

Dans 4<sup>e</sup> Dimension 2004, les paramètres d'impression courants (utilisés en mode Structure) et les paramètres d'impression des formulaires (utilisés dans les modes Utilisation et Menus créés) sont dissociés. Le bouton Paramètres/**Edition...** du thème "Impression" de la Liste des propriétés du formulaire permet d'accéder aux paramètres d'impression spécifiques du formulaire :


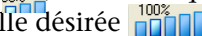




Accès aux paramètres d'impression du formulaire

Les paramètres d'impression accessibles depuis le menu **Fichier** sont utilisés pour la session courante en mode Structure et ne sont pas conservés.

## Zoom

Il est désormais possible de zoomer dans le formulaire courant.

Vous pouvez passer en mode "Zoom" soit en cliquant sur le bouton loupe  soit en cliquant directement sur la barre correspondant à l'échelle désirée  dans la barre d'outils de la fenêtre (les paliers d'affichage sont 50%, 100%, 200%, 400% et 800%).

- Lorsque vous cliquez sur le bouton loupe, le curseur prend la forme d'une loupe . Pour augmenter le pourcentage d'affichage d'un palier, cliquez dans le formulaire. Pour réduire le pourcentage d'affichage d'un palier, appuyez sur la touche **Maj** (le curseur se transforme en ) et cliquez dans le formulaire.



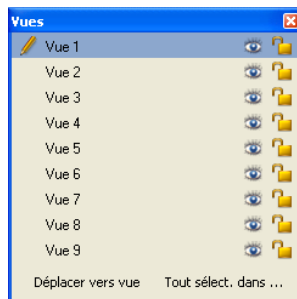
- Lorsque vous cliquez sur une barre de pourcentage, l'affichage est immédiatement modifié.


En mode Zoom, toutes les fonctions de l'éditeur de formulaires restent disponibles.

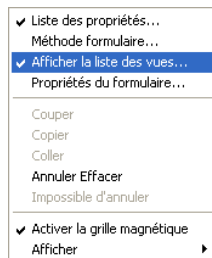
## Gestion des vues

Vous pouvez utiliser des **vues** dans les formulaires de 4<sup>e</sup> Dimension 2004. Cette nouvelle fonction facilite la construction de formulaires complexes en permettant de répartir les objets au sein de différentes vues qu'il est ensuite possible de masquer ou d'afficher en fonction des besoins. La répartition des objets peut s'effectuer par exemple en fonction de leur nature (champs, variables, objets statiques, etc.). Tous les types d'objets, y compris les sous-formulaires et les zones de plug-ins, sont inclus dans les vues.

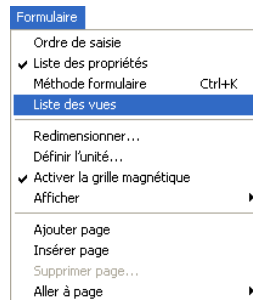
Vous disposez de 9 vues par formulaire, nommées par défaut Vue 1 à Vue 9 (leur nom peut être modifié). Chaque vue peut être affichée, masquée ou verrouillée. Leur gestion s'effectue via la palette des vues :



Pour afficher cette palette, vous pouvez cliquer sur le bouton des vues dans la barre d'outils de la fenêtre  ou bien choisir la commande **Afficher la liste des vues** dans le menu contextuel ou **Liste des vues** dans le menu **Formulaire** :



Menu contextuel

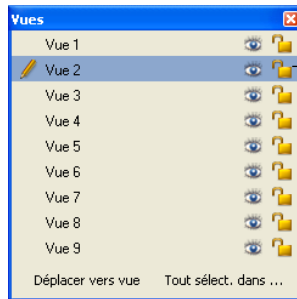


- Notes*
- Les vues sont un outil purement graphique, utilisable uniquement dans l'éditeur de formulaires ; il n'est pas possible d'accéder aux vues par programmation ou en mode Utilisation.
  - Les objets d'une même vue peuvent appartenir à des pages différentes d'un formulaire ; seuls les objets de la page courante (et de la page 0 si elle est visible) peuvent être affichés, quelle que soit la configuration des vues.
  - Les vues sont indépendantes des plans des objets, il n'y a pas de hiérarchie d'affichage entre les différentes vues.
  - Seuls les objets appartenant à la vue courante peuvent être groupés.

### Placer un objet dans une vue

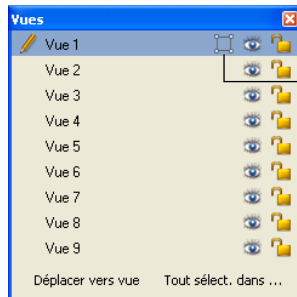
Un objet ne peut appartenir qu'à une seule vue. Tout objet créé dans un formulaire est placé dans la vue courante. Par défaut, la Vue 1 est sélectionnée, par conséquent tous les objets sont placés dans la première vue du formulaire.

Pour créer un objet dans une autre vue, il suffit de la sélectionner préalablement dans la palette en cliquant sur sa ligne :



Vue sélectionnée (tous les objets seront créés par défaut dans cette vue)

Il est également possible de déplacer un ou plusieurs objet(s) d'une vue à une autre. Pour cela, effectuez dans le formulaire la sélection d'objet(s) à changer de vue. La liste des vues indique à l'aide d'un symbole la vue à laquelle appartient la sélection :



Symbole de sélection : la sélection d'objets appartient à la vue 1

*Note* La sélection peut contenir plusieurs objets appartenant à des vues différentes.

Il vous suffit ensuite de sélectionner la vue de destination de la sélection puis de cliquer sur le bouton **Déplacer vers vue**. La sélection est alors déplacée dans la nouvelle vue :

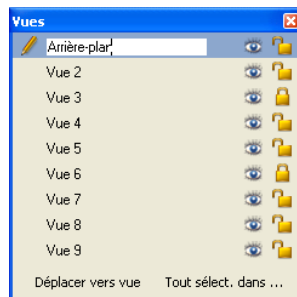


*Note* Vous pouvez afficher la vue dans laquelle se trouve chaque objet à l'aide des badges de 4<sup>e</sup> Dimension. Pour plus d'informations sur ce point, reportez-vous au [paragraphe "Utilisation des badges", page 125](#).

## Renommer une vue

Vous pouvez renommer chacune des 9 vues, par exemple pour caractériser les objets qu'elles contiennent. Vous pouvez renommer les 9 vues d'une manière différente dans chaque formulaire de la base.


Pour renommer une vue, vous pouvez soit utiliser **Ctrl+clic** (Windows) ou **Commande+clic** (MacOS) sur le nom de la vue, soit double-cliquer sur le nom de la vue (dans ce cas, la vue est sélectionnée). Le nom devient alors éditable :



## Travailler avec les vues

Une fois que vous avez réparti chaque objet dans les vues, vous pouvez utiliser la palette des vues pour :

- sélectionner en un clic tous les objets d'une même vue,


- afficher ou masquer les objets de chaque vue,
  - verrouiller les objets d'une vue.
- **Sélectionner tous les objets d'une vue**  
Vous pouvez sélectionner dans la page courante du formulaire tous les objets appartenant à une même vue. Cette fonction est utile pour appliquer des modifications globales à un ensemble d'objets. Pour cela, sélectionnez la vue de laquelle vous souhaitez sélectionner tous les objets et cliquez sur le bouton **Tout sélect. dans...**
- **Afficher ou masquer les objets d'une vue**  
Vous pouvez à tout moment afficher ou masquer les objets d'une vue dans la page courante du formulaire. Cette fonction permet par exemple de se concentrer sur certains objets lors de la modification du formulaire. Par défaut, toutes les vues sont affichées, comme l'indique l'icône  en regard de chaque vue dans la palette des vues. Pour masquer une vue, cliquez sur cette icône. Elle est alors grisée et les objets de la vue correspondante ne sont plus affichés dans le formulaire :

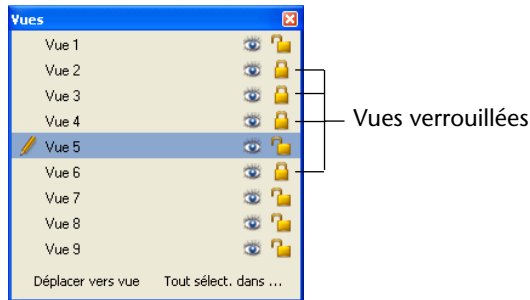


*Note* Il n'est pas possible de masquer la vue courante (sélectionnée dans la liste des vues).

Pour afficher une vue masquée, il suffit de la sélectionner ou de cliquer de nouveau sur l'icône de visualisation.

- **Verrouiller les objets d'une vue**  
Vous pouvez verrouiller les objets d'une vue, c'est-à-dire empêcher leur sélection, leur modification ou leur suppression dans le formulaire. Une fois verrouillé, un objet ne peut pas être sélectionné par un clic, un rectangle de sélection ou la commande **Sélectionner objets de même type** du menu contextuel. Cette fonction est utile pour éviter les erreurs de manipulation.

Par défaut, toutes les vues sont déverrouillées, comme l'indique l'icône  en regard de chaque vue dans la palette des vues. Pour verrouiller les objets d'une vue, cliquez sur cette icône. Le cadenas est alors refermé, ce qui indique que la vue est verrouillée :



*Note* Il n'est pas possible de verrouiller la vue courante (sélectionnée dans la liste des vues).

Pour déverrouiller une vue, il suffit de la sélectionner ou de cliquer à nouveau sur l'icône de verrouillage.

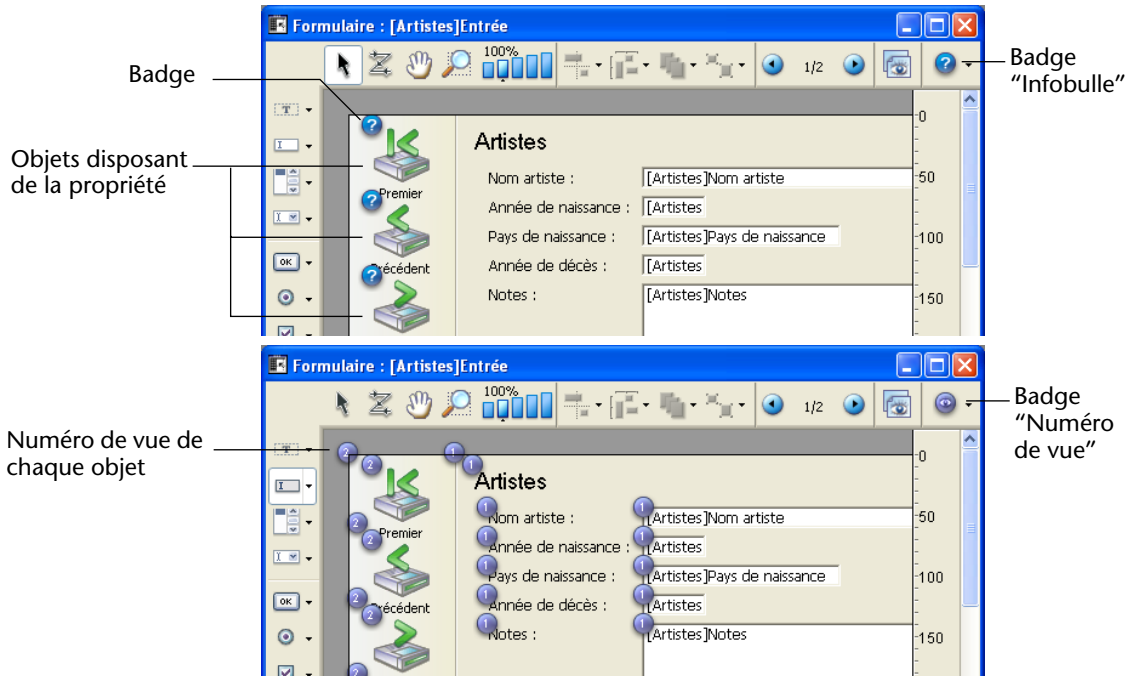
## Utilisation des badges


L'éditeur de formulaires de 4<sup>e</sup> Dimension 2004 permet d'utiliser des *badges* afin de faciliter la visualisation des propriétés des objets.

Le principe de cette fonction est simple : chaque badge est associé à une propriété (par exemple **Infobulle**, signifiant "dispose d'une infobulle associée"). Lorsque vous activez un badge, 4<sup>e</sup> Dimension affiche une petite icône (un badge) en haut à gauche de chaque objet du formulaire auquel s'applique la propriété :

Pour les propriétés de type "vrai/faux" (telles que "Infobulle"), le badge n'est affiché que si l'objet dispose de la propriété.

Pour les propriétés de type "valeur" (par exemple "Numéro de vue"), le badge affiche la valeur de la propriété pour chaque objet.




















 Pour activer un badge, cliquez une ou plusieurs fois sur le bouton de sélection des badges jusqu'à ce que le badge souhaité soit sélectionné. Vous pouvez également cliquer sur la partie droite du bouton et sélectionner directement le type de badge à afficher dans le menu associé :



Pour ne pas afficher de badges, choisissez la ligne **Pas de badges** dans le menu de sélection.

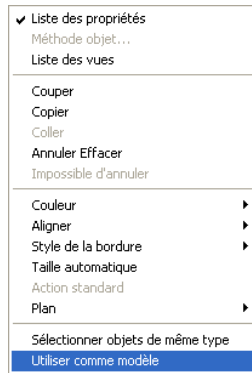
*Note* Vous pouvez définir les badges à afficher par défaut dans la page **Editeur de formulaires** (thème “Mode Structure”) des Préférences de l’application. Pour plus d’informations, reportez-vous au [paragraphe “Editeur de formulaires”, page 60](#).

Voici la description de chaque type de badge :

<b>Icône</b>	<b>Nom</b>	<b>Description</b>
	Méthode objet	Est affiché pour les objets auxquels une méthode objet est associée
	Action standard	Est affiché pour les objets auxquels une action standard est associée
	Redimensionnement	Est affiché pour les objets disposant d’au moins une propriété de redimensionnement, indique la combinaison de propriétés courante
	Agrandir horizontalement	
	Déplacer horizontalement	
	Agrandir verticalement	
	Déplacer verticalement	
	Agrandir horizontalement, Agrandir verticalement	
	Agrandir horizontalement, Déplacer verticalement	
	Déplacer horizontalement, Agrandir verticalement	
	Déplacer horizontalement, Déplacer verticalement	
	Ordre de saisie	Est affiché pour les objets saisissables, indique le numéro d’ordre de saisie
	Numéro de vue	Est affiché pour tous les objets, indique le numéro de vue
	Feuille de style	Est affiché pour les objets auxquels une feuille de style est associée
	Filtre	Est affiché pour les objets saisissables auxquels un filtre de saisie est associé
	Infobulle	Est affiché pour les objets auxquels une infobulle (message d’aide) est associée
	Pas de badge	Aucun badge n’apparaît

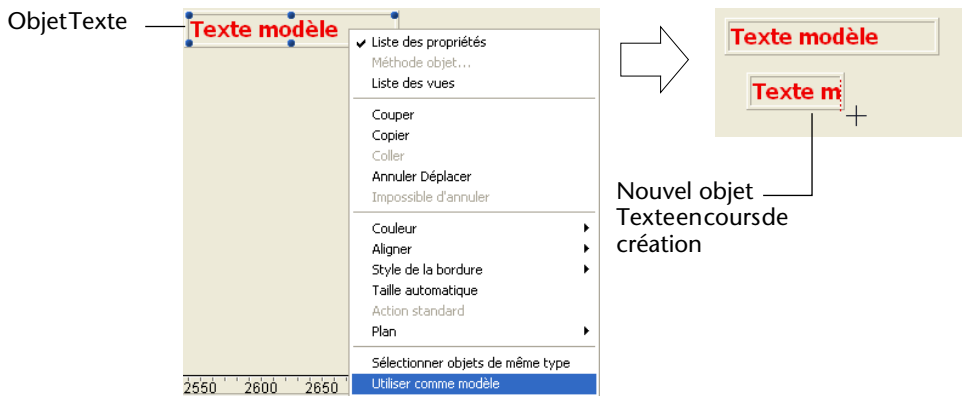
## Utiliser comme modèle

Une nouvelle commande est disponible dans le menu contextuel de l'éditeur de formulaires, lorsque le clic est effectué sur un objet :  
**Utiliser comme modèle.**



Cette commande provoque l'enregistrement comme modèle de l'objet sur lequel le clic a eu lieu ; le modèle ainsi désigné et ses propriétés seront utilisés par la suite lors de la création de tous les objets de même type.

Dans l'exemple suivant, la commande est appliquée à un objet de type Texte. Tout objet Texte créé ensuite reprend par défaut les propriétés de l'objet désigné :



Un modèle personnalisé peut être utilisé pour chaque type d'objet. Il existe un seul modèle personnalisé par type d'objet pour toute la base.

Le modèle enregistre toutes les propriétés de l'objet désigné au moment où la commande est exécutée, à l'exception des coordonnées absolues et de la méthode objet (le cas échéant).



Pour supprimer un modèle d'objet, vous devez en créer un nouveau. Vous pouvez également supprimer tous les modèles d'objets (et revenir aux modèles standard) à l'aide du bouton **Effacer tous les modèles personnalisés** dans la page **Editeur de formulaires** (thème "Mode Structure") des Préférences de l'application. Pour plus d'informations, reportez-vous au [paragraphe "Editeur de formulaires", page 60](#).

## Sélection d'objets groupés ou à l'arrière-plan

La fonction du raccourci-clavier **Ctrl+clik** (Windows) ou **Commande+clik** (MacOS) a été modifiée dans 4<sup>e</sup> Dimension 2004. Désormais, ce raccourci permet de sélectionner plus facilement des objets groupés ou situés à l'arrière-plan.

- **Objets groupés** : le raccourci **Ctrl+clik** / **Commande+clik** permet de sélectionner un objet appartenant à un groupe sans devoir le dégroupier au préalable.
- **Objets à l'arrière-plan** : lorsque plusieurs objets sont superposés, le raccourci **Ctrl+clik** / **Commande+clik** permet de sélectionner successivement chaque objet en descendant d'un plan à chaque clic.

La précédente fonction de ce raccourci (sélection de tous les objets du même type) est désormais accessible via la commande **Sélectionner objets de même type** dans le menu contextuel de l'éditeur.

## Utiliser la grille magnétique

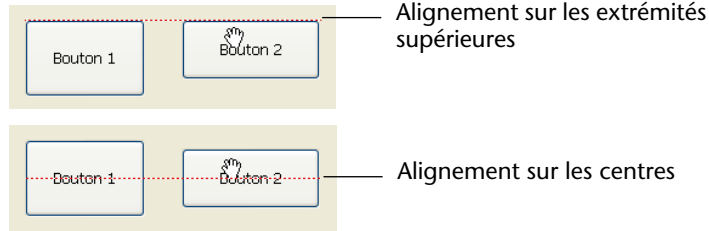
La fonction d'alignement "magnétique" des objets a été modifiée dans 4<sup>e</sup> Dimension 2004.

Désormais, l'alignement magnétique des objets est basée sur la position relative des objets entre eux. Le magnétisme n'est utilisable que lorsqu'au moins deux objets sont présents dans le formulaire. Il n'est plus possible d'afficher une grille visible en permanence dans l'éditeur de formulaires.

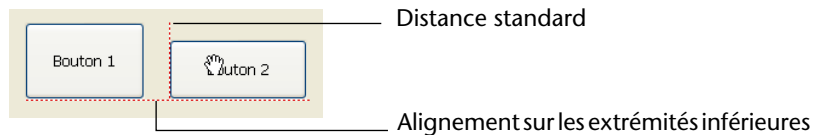
Le principe est le suivant : lorsque vous faites glisser un objet dans le formulaire, 4<sup>e</sup> Dimension indique des emplacements possibles pour cet objet sur la base d'*alignements remarquables* avec les autres objets du formulaire. Un alignement remarquable est établi à chaque fois que :

- horizontalement, les extrémités ou les centres de deux objets coïncident,
- verticalement, les extrémités de deux objets coïncident.

A ce moment, 4<sup>e</sup> Dimension place l'objet à l'emplacement et affiche un trait rouge indiquant l'alignement remarquable pris en compte :



En ce qui concerne la répartition des objets, 4<sup>e</sup> Dimension propose une distance basée sur les standards d'interface. Comme pour l'alignement magnétique, des traits rouges indiquent les distances remarquables au moment où elles sont atteintes.



Enfin, le magnétisme entraîne également l'observation de paliers lors du redimensionnement manuel des objets.

Ces nouveaux mécanismes permettent de construire rapidement des interfaces fonctionnelles. Bien entendu, ils s'appliquent à tous les types d'objets des formulaires. Comme dans les versions précédentes de 4<sup>e</sup> Dimension, le magnétisme peut être activé ou désactivé à tout moment à l'aide de la commande **Activer la grille magnétique** du menu **Formulaire** ou du menu contextuel de l'éditeur. Il est également possible de définir l'activation par défaut de cette fonction dans la page **Editeur de formulaires** (thème "Mode Structure") des Préférences de l'application. Pour plus d'informations, reportez-vous au [paragraphe "Editeur de formulaires", page 60](#).

---

*Note* Il est possible d'activer ou de désactiver temporairement la grille magnétique lorsqu'un objet est sélectionné en appuyant sur la touche **Ctrl** (Windows) ou **Control** (MacOS).

---

### Modifiable par l'utilisateur

La nouvelle propriété **Modifiable par l'utilisateur** permet d'indiquer que le formulaire peut être affiché dans l'éditeur de formulaires utilisateurs (cf. [paragraphe "Editeur de formulaires utilisateurs", page 29](#)).

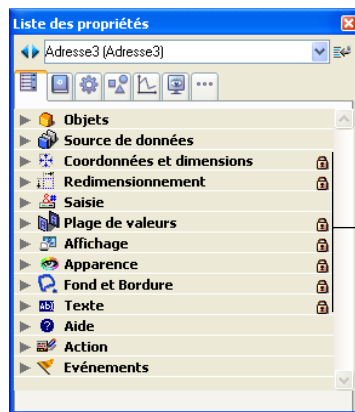
Lorsque cette propriété est cochée, le formulaire est automatiquement verrouillé au moment de sa sauvegarde. Si vous l'ouvrez par la suite sans le déverrouiller, aucune action autre que la sélection n'est autorisée (un bip retentit). Vous devrez explicitement déverrouiller le formulaire en cliquant sur l'icône en forme de cadenas afin de pouvoir le modifier. A noter que le déverrouillage d'un formulaire rend obsolètes les formulaires utilisateurs éventuellement définis. Pour plus d'informations sur ce mécanisme, reportez-vous au [paragraphe "Formulaires utilisateurs"](#), page 254.



Une fois qu'un formulaire "Modifiable par l'utilisateur" est déverrouillé, vous pouvez désigner les propriétés du formulaire ou de chaque objet qui seront modifiables dans l'éditeur de formulaires utilisateurs. Ce paramétrage est effectué dans la Liste des propriétés, au niveau de chaque thème de propriétés.

Seules certaines propriétés peuvent être déclarées modifiables. En particulier, toutes les propriétés liés à la programmation (méthode, événements...) ne sont jamais modifiables.

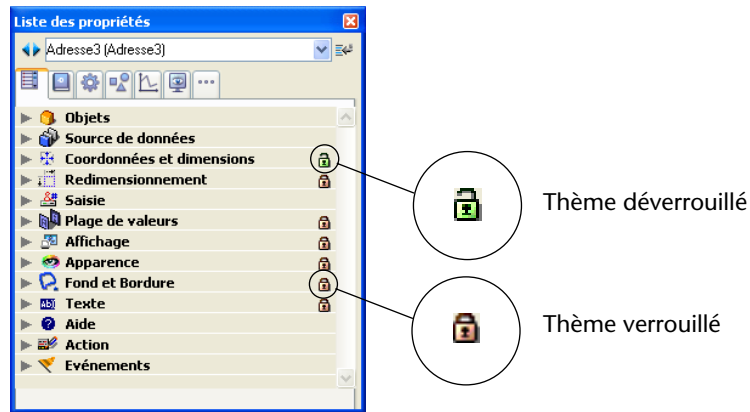
Les propriétés modifiables sont repérées via une icône de cadenas dans la Liste des propriétés. Par défaut, toutes ces propriétés sont verrouillées (aucune propriété n'est modifiable par l'utilisateur) :



Thèmes pouvant être déclarés modifiables

Pour déverrouiller un thème de propriétés (et donc autoriser sa modification dans l'éditeur de formulaires utilisateurs), il suffit de cliquer sur le cadenas.

La même opération permet de le verrouiller à nouveau :



## Ajustement dynamique

La propriété de formulaire **Ajustement dynamique** permet de modifier le mode de redimensionnement des formulaires. Cette propriété est disponible dans le thème "Taille du formulaire" pour les formulaires convertis depuis une version précédente de 4<sup>e</sup> Dimension, lorsque la propriété **Taille auto** est sélectionnée.

Dans les versions précédentes de 4<sup>e</sup> Dimension, lorsqu'un formulaire disposait de la propriété **Taille auto**, la taille du formulaire était calculée à l'ouverture du formulaire uniquement. Si des modifications étaient effectuées par la suite à l'aide de commandes du type DEPLACER OBJET, la taille du formulaire n'était pas mise à jour.

Avec 4<sup>e</sup> Dimension 2004, la taille du formulaire est dynamiquement mise à jour dans ce cas.

Ce principe est activé pour les formulaires créés en version 2004. En revanche, pour des raisons de compatibilité les formulaires des bases converties n'en bénéficient pas par défaut. Si vous souhaitez que la taille des formulaires convertis soit dynamiquement mise à jour, cochez l'option **Ajustement dynamique**.

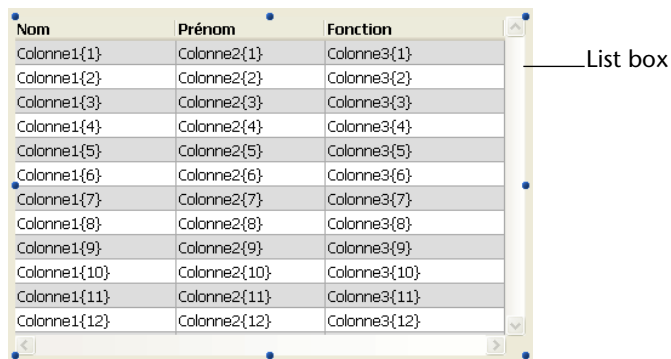
## Objets de formulaires

De nombreuses nouveautés ont été apportées aux objets de formulaires dans 4<sup>e</sup> Dimension 2004 :

- introduction d'un nouveau type d'objet : List Box
- meilleure prise en charge des séparateurs
- nouveaux principes de gestion de l'interface de plate-forme
- utilisation de contrôles de saisie natifs
- correction orthographique pour les champs et les variables
- cases à cocher à trois états
- nouveaux boutons 3D
- enfin, divers fonctionnements ont été étendus ou améliorés : type booléen pour les cases à cocher, mise à jour des références incluses, nouvelle propriété Invisible par défaut, groupement des boutons radio.

### List Box

4<sup>e</sup> Dimension 2004 propose un nouveau type d'objet de formulaire : les List Box.

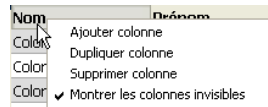


Les list box sont des zones permettant d'afficher et de saisir des données sous forme de colonnes synchronisées. Ces objets sont semblables aux "zones de défilement groupées" des versions précédentes de 4<sup>e</sup> Dimension, dont ils reprennent et étendent les possibilités (saisie de valeurs, tri des colonnes, apparence personnalisée, glisser-déposer, etc.). Un objet de type List box est entièrement paramétrable dans l'éditeur de formulaires de 4<sup>e</sup> Dimension et peut également être contrôlé par programmation.

Ce paragraphe détaille les nouveautés liées à la création et au paramétrage des objets de type List box dans l'éditeur de formulaires. Pour plus d'informations sur la gestion programmée de ces objets, reportez-vous au [paragraphe "List box", page 328](#).

### Principales caractéristiques

Une list box contient une ou plusieurs colonnes dont le contenu est automatiquement synchronisé. Par défaut lorsque vous créez une list box, elle contient une seule colonne. Vous pouvez modifier le nombre de colonnes (ajouter, dupliquer ou supprimer une colonne) à l'aide du menu contextuel (clic sur une colonne ou un en-tête de colonne) ou dans les propriétés de la list box.

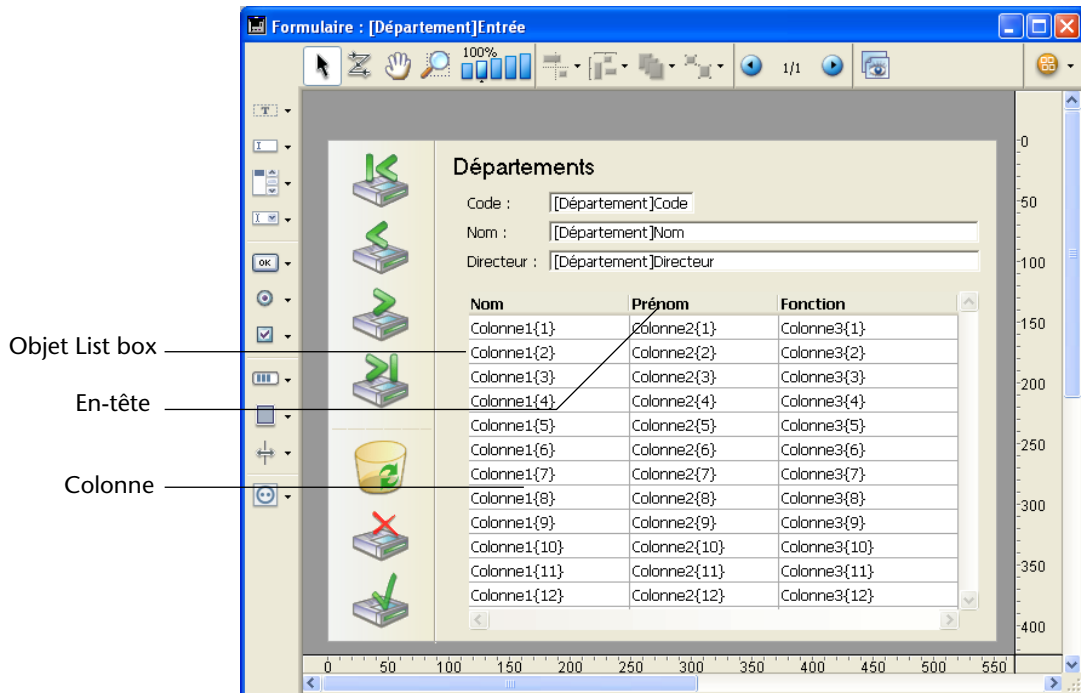


Le nombre de colonnes est en principe illimité (il dépend des ressources de la machine).

Chaque colonne de la list box est associée à un tableau 4D. Par défaut, 4<sup>e</sup> Dimension affecte le nom "ColonneN" à chaque variable de colonne, et donc à chaque tableau associé. Vous pouvez le modifier dans les propriétés des colonnes. La gestion des valeurs saisies ou affichées dans les list box s'effectue donc via le langage de 4<sup>e</sup> Dimension. Pour plus d'informations sur ce point, reportez-vous au [paragraphe "Principes de programmation des objets de type List box", page 329](#). Vous pouvez associer une énumération à une colonne afin de contrôler la saisie (cf. [paragraphe "Propriétés spécifiques des colonnes de list box", page 141](#)).

Une list box est composée de trois parties distinctes : l'**objet** dans son ensemble, les **colonnes** et les **en-têtes** des colonnes. Dans l'éditeur de formulaires, ces parties peuvent être sélectionnées séparément.

Chacune dispose de son propre nom d'objet et de variable ainsi que de propriétés spécifiques :



Par exemple, le nombre de colonnes ou la couleur alternée de chaque ligne sont définies dans les propriétés de l'objet list box, la largeur de chaque colonne est définie dans les propriétés des colonnes et la police de l'en-tête est définie dans les propriétés des en-têtes.

### En mode Utilisation

En mode Utilisation, les list box permettent d'afficher et de saisir des données sous forme de listes.

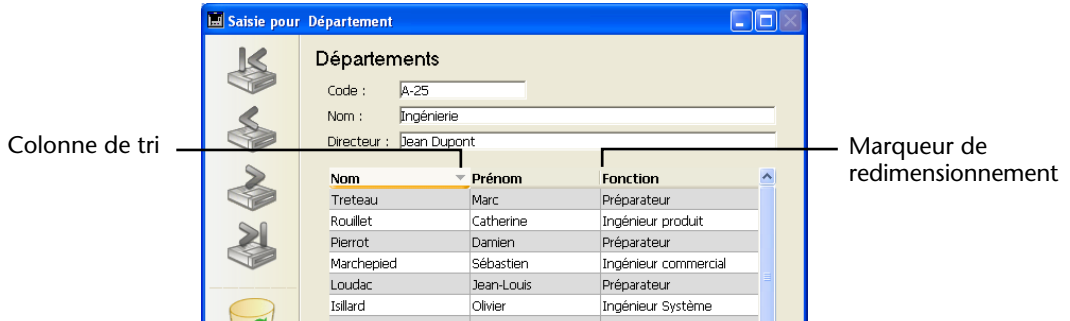
Pour passer une cellule en mode édition (si la saisie est autorisée pour la colonne associée), il suffit de cliquer deux fois sur la valeur qu'elle contient :

Nom	Prénom
Dupont	Jean
Durant	Marc

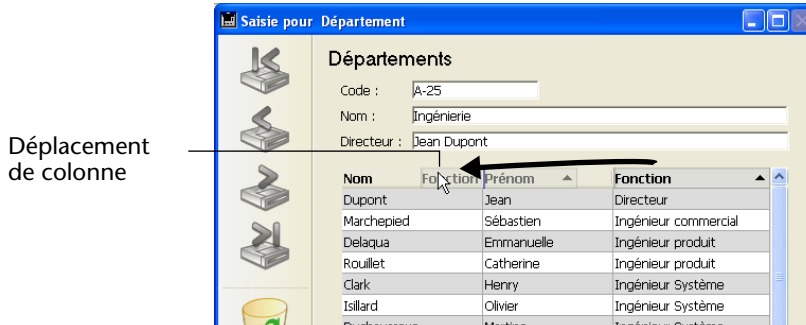
Il est possible de trier les valeurs des colonnes via un clic sur un en-tête. Le tri est alphanumérique, alternativement croissant / décroissant lors de clics multiples. Toutes les colonnes sont toujours automatiquement

synchronisées.

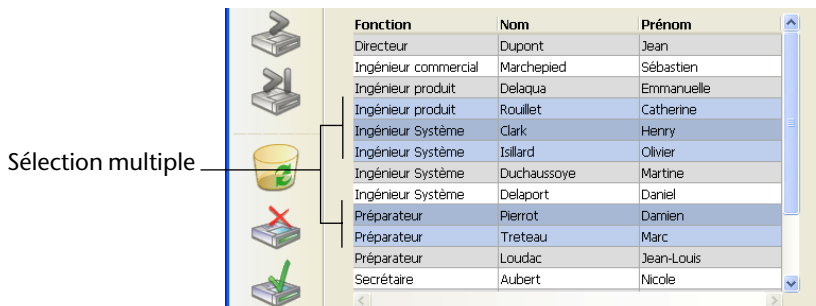
Il est également possible de redimensionner chaque colonne :



L'utilisateur peut modifier l'ordre des colonnes et des lignes en les déplaçant à l'aide de la souris (si cette action est autorisée) :



Enfin, l'utilisateur peut sélectionner une ou plusieurs lignes à l'aide des raccourcis standard : **Maj+clik** pour une sélection continue et **Ctrl+clik** (Windows) ou **Commande+clik** (MacOS) pour une sélection discontinue.

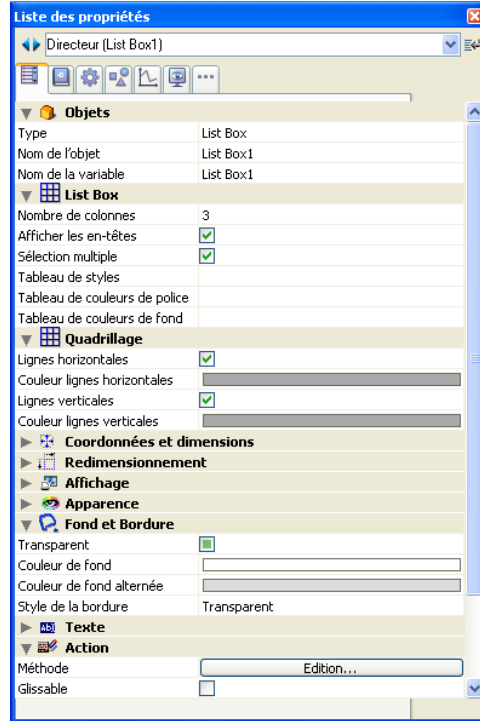


Toutes ces caractéristiques peuvent être contrôlées via les propriétés de la list box, des colonnes et des en-têtes. Elles sont détaillées dans les paragraphes suivants.



## Propriétés spécifiques de l'objet List Box

Lorsque vous sélectionnez un objet List Box dans l'éditeur de formulaires, la liste des propriétés apparaît ainsi :

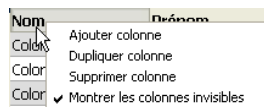


Cette section décrit les propriétés spécifiques de l'objet List Box.

### ■ Thème List Box

Ce thème regroupe les propriétés élémentaires de l'objet.

- **Nombre de colonnes** : nombre de colonnes affichées dans la list box (1 par défaut). Vous pouvez modifier le nombre de colonnes (ajouter, dupliquer ou supprimer une colonne) à l'aide du menu contextuel (clic sur une colonne ou un en-tête de colonne) :



- **Nombre de colonnes fixes** : nombre de colonnes dont le déplacement est interdit en mode Utilisation/Menus créés. La valeur saisie indique le nombre de colonnes fixes à partir de la première colonne située à gauche. Pour interdire tout déplacement de colonnes, la valeur saisie doit être égale au nombre total de colonnes dans la list box.

La propriété tient compte des colonnes invisibles.

Par défaut, la valeur de la propriété est 0, ce qui signifie que toutes les colonnes sont déplaçables.

La nouvelle commande **NUMERO COLONNE LISTBOX DEPLACEE** permet de gérer le déplacement des colonnes de list box.

- **Afficher les en-têtes** : permet d'afficher ou de masquer les en-têtes des colonnes (affichés par défaut).
- **Sélection multiple** : permet d'autoriser ou d'interdire les sélections multiples de lignes. Par défaut, les sélections multiples peuvent être effectuées à l'aide des raccourcis standard : **Maj+clic** pour une sélection continue et **Ctrl+clic** (Windows) ou **Commande+clic** (MacOS) pour une sélection discontinue.  
Si cette option est désélectionnée, seule la sélection d'une ligne à la fois sera possible dans la list box.
- **Tableau de styles** : permet de saisir le nom d'un tableau Entier long utilisé pour appliquer un style de caractères personnalisé à chaque ligne de la list box. Chaque élément de ce tableau correspondra à une ligne de la list box, le tableau doit donc avoir la même taille que les tableaux associés aux colonnes.  
Pour remplir le tableau (à l'aide d'une méthode), utilisez les constantes du thème "Styles de caractères". En additionnant les constantes, vous pouvez combiner les styles.  
Pour appliquer le style défini dans les propriétés de la list box à une ligne, passez la valeur -255 à l'élément du tableau correspondant.
- **Tableau de couleurs de police** : permet de saisir le nom d'un tableau Entier long utilisé pour appliquer une couleur de police de caractères personnalisée à chaque ligne de la list box. Chaque élément de ce tableau correspondra à une ligne de la list box, le tableau doit donc avoir la même taille que les tableaux associés aux colonnes.  
Vous devez remplir le tableau (à l'aide d'une méthode) avec des valeurs de couleurs RVB. Pour plus d'informations sur ce point, reportez-vous à la description de la commande **FIXER COULEURS RVB**.  
Pour appliquer la couleur de police définie dans les propriétés de la list box à une ligne, passez la valeur -255 à l'élément du tableau correspondant.
- **Tableau de couleurs de fond** : permet de saisir le nom d'un tableau Entier long utilisé pour appliquer une couleur de fond personnalisée à chaque ligne de la list box. Chaque élément de ce tableau correspondra à une ligne de la list box, le tableau doit donc avoir la même

taille que les tableaux associés aux colonnes.

Vous devez remplir le tableau (à l'aide d'une méthode) avec des valeurs de couleurs RVB. Pour plus d'informations sur ce point, reportez-vous à la description de la commande `FIXER COULEURS RVB`.

Pour appliquer la couleur de fond définie dans les propriétés de la list box à une ligne, passez la valeur -255 à l'élément du tableau correspondant.

---

*Note* Les tableaux de styles, de couleurs de police et de couleurs de fond sont retournés par la commande `LIRE TABLEAUX LISTBOX`.

---

#### ■ **Thème Quadrillage**

Ce thème regroupe les propriétés liées à la grille affichée dans l'objet list box.

- **Lignes horizontales** : affiche ou masque les lignes horizontales de la list box (affichées par défaut).
- **Couleur lignes horizontales** : définit la couleur des lignes horizontales de la list box (gris par défaut).
- **Lignes verticales** : affiche ou masque les lignes verticales de la list box (affichées par défaut).
- **Couleur lignes verticales** : définit la couleur des lignes verticales de la list box (gris par défaut).

#### ■ **Thème Coordonnées et dimensions**

Ce thème regroupe les propriétés standard relatives aux coordonnées, à la largeur et à la hauteur de la list box.

La propriété **Hauteur des lignes** est spécifique : elle vous permet de définir la hauteur des lignes de la list box (en pixels). Cette hauteur est également appliquée aux en-têtes. La hauteur des lignes est définie par défaut en fonction de la plate-forme et de la taille de police.

#### ■ **Thème Redimensionnement**

Ce thème regroupe les propriétés standard relatives au comportement de la list box en cas de redimensionnement du formulaire.

#### ■ **Thème Affichage**

Ce thème regroupe les propriétés standard relatives à la visibilité de la list box dans le formulaire. Pour plus d'informations sur la propriété **Invisible par défaut**, reportez-vous au [paragraphe "Nouvelle propriété Invisible par défaut"](#), page 165.

### ■ **Thème Apparence**

Ce thème regroupe les propriétés standard relatives à l'interface de plate-forme de la list box. Vous pouvez également spécifier dans ce thème si vous souhaitez que la list box dispose ou non de barres de défilement horizontale et verticale.

### ■ **Thème Fond et Bordure**

Ce thème regroupe les propriétés relatives aux couleurs de fond des lignes ainsi qu'au style de la bordure.

■ **Couleur de fond** : permet de définir la couleur de fond de la list box. Cette couleur est utilisée pour la totalité de l'objet à l'exception des en-têtes (s'ils sont affichés).

■ **Couleur de fond alternée** : permet de définir une couleur de fond différente pour les lignes impaires de la list box. Utiliser une couleur de fond alternée facilite la lecture des tableaux.

■ **Style de la bordure** : permet de définir un style standard pour la bordure (le contour) de l'objet list box.

### ■ **Thème Texte**

Ce thème regroupe les propriétés standard relatives aux textes affichés dans la list box (feuille de style, police, attributs, etc.). A noter que des propriétés de texte spécifiques peuvent être définies au niveau de chaque colonne et en-tête.

### ■ **Thème Action**

Ce thème regroupe les propriétés relatives au comportement dynamique de la list box.

■ **Méthode (Edition...)** : ce bouton affiche la méthode de l'objet list box (à noter que chaque colonne peut également comporter une méthode objet).

■ **Lignes déplaçables** : autorise le déplacement de lignes en mode Utilisation/Menus créés. Cette option est cochée par défaut. La nouvelle commande **NUMERO LIGNE LISTBOX DEPLACEE** permet de gérer le déplacement des lignes de list box.

■ **Triable** : permet le tri des données des colonnes via un clic sur leur en-tête. Cette option est cochée par défaut. Les tableaux (colonnes) de type image ne peuvent pas être triés via ce mécanisme.

### ■ **Thème Événements**

Ce thème regroupe les événements formulaire associés à la list box. De nouveaux événements ont été ajoutés dans 4<sup>e</sup> Dimension 2004 afin de

prendre en charge les spécificités de programmation des list box. Pour plus d'informations, reportez-vous au [paragraphe "Événements formulaire liés aux list box"](#), page 304.

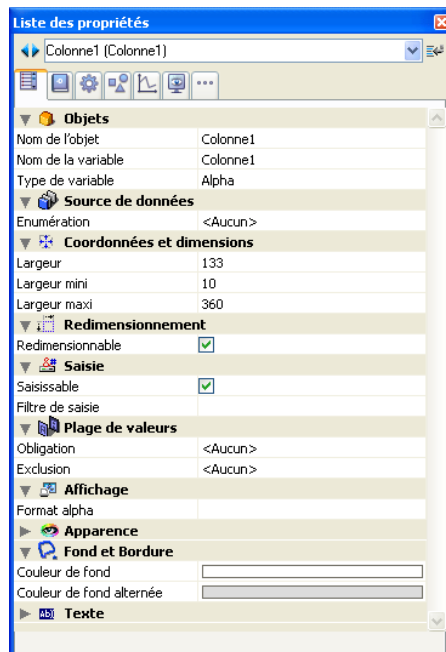
### Propriétés spécifiques des colonnes de list box

Vous pouvez sélectionner une colonne de list box dans l'éditeur de formulaires en cliquant dessus lorsque l'objet List box est sélectionné :

Nom	Prénom	Fonction
Colonne1{1}	Colonne2{1}	Colonne3{1}
Colonne1{2}	Colonne2{2}	Colonne3{2}
Colonne1{3}	Colonne2{3}	Colonne3{3}
Colonne1{4}	Colonne2{4}	Colonne3{4}
Colonne1{5}	Colonne2{5}	Colonne3{5}
Colonne1{6}	Colonne2{6}	Colonne3{6}

Colonne sélectionnée

Dans ce cas, la liste des propriétés apparaît ainsi :



Cette section décrit les propriétés spécifiques des colonnes de list box.

#### ■ Thème Source de données

Ce thème comporte une propriété permettant d'associer une **énumération** à la colonne de list box.

Si vous désignez une énumération, l'utilisateur pourra utiliser ses valeurs (affichées via un pop up menu) pour modifier les valeurs de la

colonne et de son tableau associé :

Icône d'affichage du pop up menu

Énumération

Fonction	Nom	▲ Prénom
Secrétaire	Aubert	Nicole
Technicien	Brodin	Hector
Secrétaire	Clark	Henry
Secrétaire	Delaflotte	Cédric
Préparateur	Delaport	Daniel
Ingénieur produit	Delaqua	Emmanuelle
Directeur	Duchaussoye	Martine
Ingénieur commercial	Dupont	Jean
Ingénieur Système	Durant	Marc
Technicien	Guichard	Elise
Ingénieur Système	Isillard	Olivier
Préparateur	Loudac	Jean-Louis

A noter que la saisie au clavier reste possible. Si vous souhaitez que les valeurs ne puissent être modifiées que via le pop up menu de sélection, choisissez une énumération obligatoire (cf. ci-dessous).

*Note* Si l'énumération désignée est hiérarchique, seuls les éléments de premier niveau sont pris en compte.

■ **Thème Coordonnées et dimensions**

Ce thème regroupe les propriétés liées à la largeur de la colonne de list box.

- **Largeur** : largeur par défaut de la colonne (en pixels). Cette valeur est mise à jour lorsque vous redimensionnez la colonne à l'aide de la souris dans l'éditeur de formulaires.

Prénom	Fonction
Colonne2{1}	Colonne3{1}
Colonne2{2}	Colonne3{2}

Curseur de redimensionnement

Si la propriété **Redimensionnable** est cochée (cf. ci-dessous), l'utilisateur pourra également redimensionner manuellement la colonne.

- **Largeur mini** : largeur minimale de la colonne (en pixels). La largeur de la colonne ne pourra pas être réduite au-dessous de cette valeur en cas de redimensionnement de la colonne ou du formulaire.
- **Largeur maxi** : largeur maximale de la colonne (en pixels). La colonne ne pourra pas être élargie au-delà de cette valeur en cas de redimensionnement de la colonne ou du formulaire.

*Note* En cas de redimensionnement du formulaire, si la propriété de redimensionnement horizontal **Agrandir** a été affectée à la list box, la colonne la plus à droite sera étirée au-delà de la largeur maximale si nécessaire.

---

### ■ **Thème Redimensionnement**

Ce thème comporte uniquement l'option **Redimensionnable** (cochée par défaut). Lorsque cette option est cochée, l'utilisateur peut redimensionner la colonne en déplaçant les parois de la zone d'en-tête.

### ■ **Thème Saisie**

Ce thème regroupe les propriétés liées à la saisie de données dans la colonne de list box.

■ **Saisissable** : autorise la saisie dans la colonne (cochée par défaut). Pour éditer la valeur d'une cellule, l'utilisateur doit cliquer deux fois sur la valeur.

Lorsque cette propriété est désélectionnée, les pop up menus éventuellement associés à la colonne via une énumération (thèmes "Source de données" et "Plage de valeurs") sont désactivés.

■ **Filtre de saisie** : associe un filtre de saisie aux cellules de la colonne. Cette propriété n'est pas accessible si la propriété **Saisissable** est désélectionnée.

### ■ **Thème Plage de valeurs**

Ce thème permet de désigner des énumérations utilisées pour le contrôle de la saisie dans la colonne de list box.

■ **Obligation** : permet de désigner une énumération dont seules les valeurs pourront être insérées dans la colonne. Les valeurs de l'énumération sont accessibles via un pop up menu associé à chaque cellule (cf. propriété "Énumération" du thème **Source de données**). A la différence de la propriété **Énumération**, lorsqu'une énumération obligatoire est définie, la saisie au clavier n'est plus possible, seule la sélection d'une valeur de l'énumération via le pop up menu est autorisée. Si des énumérations différentes sont définies via les propriétés **Énumération** et **Obligation**, la propriété **Obligation** a priorité.

■ **Exclusion** : permet de désigner une énumération dont les valeurs ne pourront pas être saisies dans la colonne. Si une valeur exclue est saisie, elle n'est pas acceptée et un message d'erreur est affiché.

---

*Note* Si l'énumération désignée est hiérarchique, seuls les éléments de premier niveau sont pris en compte.

---

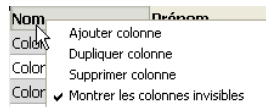
### ■ Thème Affichage

Ce thème permet de spécifier le format d’affichage des valeurs de la colonne et la propriété **Invisible**.

- Les propriétés proposées dans le menu **Format type** sont automatiquement mises à jour en fonction du type de variable défini dans le thème **Objets**. Les formats standard de 4<sup>e</sup> Dimension sont utilisables pour les données de type Alpha, Numérique, Date, Heure, Image et Booléen. Le type Texte ne dispose pas de formats d’affichage spécifique. Les formats personnalisés éventuellement créés sont également disponibles.

*Note* Les tableaux de booléens peuvent être affichés sous forme de cases à cocher ou de pop-up menus. Si vous choisissez l’option **Case à cocher**, la propriété **Titre** permet de saisir le libellé de la case à cocher. Si vous choisissez l’option **Pop-up**, les propriétés **Texte si Vrai** et **Texte si Faux** apparaissent, permettant de définir les deux libellés des pop-up menus.

- La propriété **Invisible**, lorsqu’elle est cochée, permet de rendre la colonne invisible dans les modes Utilisation et Menus créés. En mode Structure, vous avez la possibilité d’afficher ou de masquer les colonnes déclarées invisibles à l’aide du menu contextuel associé à la list box :



### ■ Thème Fond et Bordure

Ce thème regroupe les propriétés relatives aux couleurs de fond personnalisées des lignes de la colonne. Par défaut, les couleurs de fond générales de la list box sont utilisées.

- **Couleur de fond** : permet de définir une couleur de fond spécifique pour la colonne.
- **Couleur de fond alternée** : permet de définir une couleur de fond alternée spécifique pour les lignes impaires de la colonne.

### ■ Thème Texte

Ce thème regroupe les propriétés spécifiques relatives aux textes affichés dans la colonne (feuille de style, police, attributs, etc.). Par défaut, les propriétés de texte générales de la list box sont utilisées.



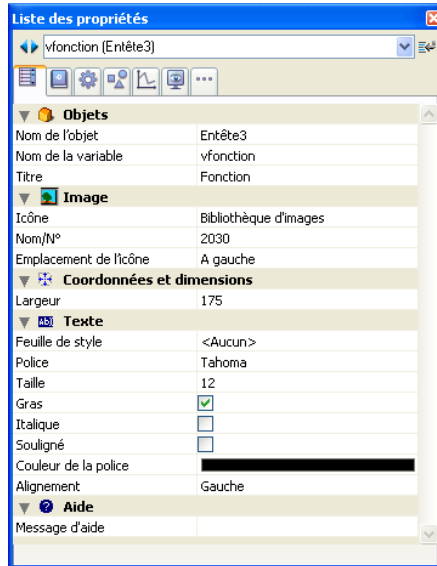
## Propriétés spécifiques des en-têtes de list box

Vous pouvez sélectionner un en-tête de list box dans l'éditeur de formulaires en cliquant dessus lorsque l'objet List box est sélectionné :

Nom	Prénom	Fonction
Colonne1{1}	Colonne2{1}	Colonne3{1}
Colonne1{2}	Colonne2{2}	Colonne3{2}
Colonne1{3}	Colonne2{3}	Colonne3{3}
Colonne1{4}	Colonne2{4}	Colonne3{4}
Colonne1{5}	Colonne2{5}	Colonne3{5}

En-tête sélectionné

Dans ce cas, la liste des propriétés apparaît ainsi :



Cette section décrit les propriétés spécifiques des en-têtes de list box.

### ■ Thème Objets

Ce thème comporte les propriétés de définition de l'en-tête.

- **Nom de l'objet** : nom de l'objet En-tête.

- **Nom de la variable** : nom de la variable associée à l'objet En-tête. Cette variable (numérique) permet de gérer par programmation le tri courant de la colonne et l'affichage de la flèche de tri (cf. [paragraphe "Principes de programmation des objets de type List box"](#), page 329).

- **Titre** : libellé apparaissant dans l'en-tête.

### ■ Thème Image

Ce thème comporte des propriétés permettant d'afficher une image dans l'en-tête de la colonne (facultatif). Une icône peut en effet être

affichée dans l'en-tête à côté ou à la place du libellé de la colonne, notamment dans le cadre des tris personnalisés.

Fonction	Nom
Directeur	Dupont
Technicien	Durant

- **Icône**  : définit la source de l'image à insérer dans l'en-tête. Comme pour les boutons image de 4<sup>e</sup> Dimension, vous pouvez utiliser une image provenant d'une variable, de la bibliothèque d'images ou d'un fichier de ressources.
- **Nom/N°**  : permet de désigner l'image à utiliser dans la source auparavant définie. Les informations à saisir dans ce champ dépendent du type de source choisi.
- **Emplacement de l'icône**  : position de l'icône dans l'en-tête. Vous pouvez la placer **A gauche** ou **A droite** de l'en-tête.
- **Thème Coordonnées et dimensions**   
Ce thème comporte la propriété **Largeur**, dont la valeur est identique par défaut à celle définie pour la colonne.
- **Thème Texte**   
Ce thème regroupe les propriétés spécifiques relatives aux textes affichés dans l'en-tête (feuille de style, police, attributs, alignement, etc.). Par défaut, les propriétés de texte générales de la list box sont utilisées.
- **Thème Aide**   
Ce thème comporte la propriété **Infobulle**, permettant d'associer une infobulle à l'en-tête. Si vous désignez un message d'aide, son contenu sera affiché lorsque l'utilisateur placera le curseur de la souris au-dessus de l'en-tête :

---

*Note* Pour plus d'informations sur la définition des messages d'aide, reportez-vous au [paragraphe "Infobulles", page 84](#).

---

## Gestion des séparateurs

La gestion des séparateurs a été modifiée dans 4<sup>e</sup> Dimension 2004 afin de rendre leur utilisation plus aisée et conforme aux standards d'interface. Les modifications concernent les aspects suivants :

- Nouvelle propriété **Pousseur**
- Gestion de la "zone sensible"
- Prise en charge des séparateurs affichés en page 0
- Affichage en temps réel et déclenchement de l'événement formulaire Sur clic.

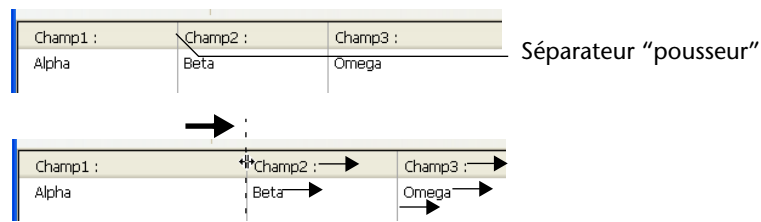
*Note* Les parois des fenêtres se comportent également comme des séparateurs pousseurs. Pour plus d'informations sur ce point, reportez-vous au [paragraphe "Bords pousseurs", page 283](#).

## Propriété Pousseur

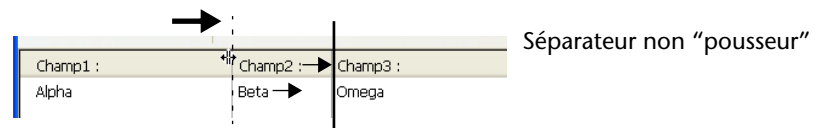
Une nouvelle propriété est disponible pour les objets séparateurs dans le thème "Redimensionnement" de la Liste des propriétés : **Pousseur**.

Lorsqu'un objet séparateur dispose de cette propriété, les autres objets situés à sa droite (séparateur vertical) ou au-dessous (séparateur horizontal) sont déplacés en même temps que lui : ils sont "poussés" — ou "tirés".

Voici le résultat du déplacement d'un séparateur "pousseur" :



Lorsque cette propriété n'est pas appliquée au séparateur, le résultat est le suivant :



Cette propriété est cochée par défaut pour les nouvelles bases et désélectionnée dans les formulaires des bases converties.

## Zone sensible

La gestion de la "zone sensible" des séparateurs, c'est-à-dire la zone dans laquelle l'utilisateur peut cliquer pour activer le séparateur, a été optimisée, notamment pour les objets séparateurs dont la largeur est très réduite (par exemple 1 pixel).

## Séparateurs en page 0

Les séparateurs placés en page 0 des formulaires sont désormais actifs pour toutes les pages. Lorsqu'un séparateur en page 0 est déplacé dans la page N du formulaire, son déplacement est pris en compte dans les autres pages lorsqu'elles sont affichées, par exemple via des onglets.

### Affichage en temps réel et événement Sur clic

Lors de l'utilisation des séparateurs, le déplacement des objets est désormais affiché en temps réel.

A la différence des versions précédentes de 4<sup>e</sup> Dimension, l'événement Sur clic est maintenant généré de manière continue durant tout le déplacement.

### Interface de plate-forme

La gestion de l'interface de plate-forme a été simplifiée dans 4<sup>e</sup> Dimension 2004. Le précédent mode de gestion est toutefois maintenu dans les bases de données converties.

Deux nouvelles propriétés d'interface sont disponibles : **Système** et **Impression**. Ces nouvelles propriétés correspondent à une approche plus simple de la gestion de l'interface des formulaires : lorsqu'il est affiché à l'écran, un formulaire doit respecter l'interface de la plate-forme courante (option **Système**). Lorsqu'il est destiné à l'impression, l'apparence des objets doit être adaptée (option **Impression**), quelle que soit la plate-forme.

### Système

Cette nouvelle propriété permet d'adapter automatiquement l'apparence du formulaire ou de l'objet en fonction de la plate-forme courante d'exécution de 4<sup>e</sup> Dimension :

- lorsque le formulaire est affiché sous MacOS, le formulaire ou l'objet prend l'apparence MacOS X,
- lorsque le formulaire est affiché sous Windows :
  - sous Windows 2000, le formulaire ou l'objet prend l'apparence Windows 2000,
  - sous Windows XP, le formulaire ou l'objet prend l'apparence Windows XP ou Windows Classique en fonction du paramétrage "Apparence" courant des propriétés d'affichage du poste.

Ce mode est utilisé par défaut pour tous les formulaires et objets dans les bases créées avec 4<sup>e</sup> Dimension version 2004.

---

*Note* Dans le cadre d'une meilleure prise en charge des spécificités de la plate-forme Windows XP, de nouvelles constantes de couleur sont disponibles pour la commande **FIXER COULEURS RVB**.

---

## Impression

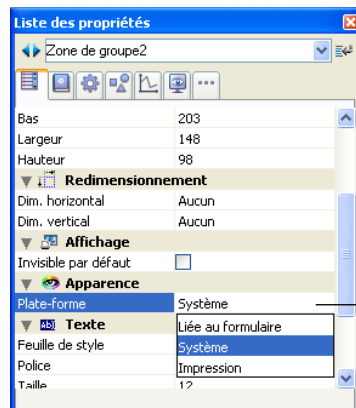
Cette nouvelle propriété permet d'adapter l'apparence du formulaire ou de l'objet en vue de l'impression : tout objet ou contrôle graphique (bouton, case à cocher, onglet, trait, etc.) est dessiné en mode vectoriel afin de produire un rendu satisfaisant à l'impression.

Le même résultat est obtenu quelle que soit la plate-forme sur laquelle le formulaire est affiché ou imprimé.

## Compatibilité et accès aux propriétés

Des propriétés différentes sont disponibles en fonction de la provenance (création ou conversion) des bases ouvertes.

- Dans les nouvelles bases de données (créées avec 4<sup>e</sup> Dimension version 2004 et suivantes), seules les propriétés de plate-forme **Système** et **Impression** sont disponibles. Elles remplacent toutes les options proposées dans les versions précédentes de 4<sup>e</sup> Dimension. L'option **Système** est appliquée par défaut à tous les formulaires et objets de la base. Elle peut être modifiée uniquement au niveau de chaque formulaire ou objet.

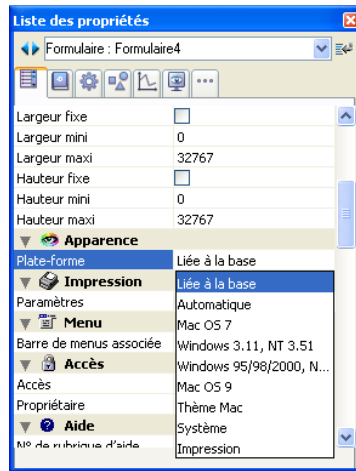


Options de plate-forme pour un objet

Enfin, les commandes FIXER INTERFACE et Lire interface n'ont pas d'effet et ne doivent plus être utilisées.

- Dans les bases de données converties depuis des versions précédentes de 4<sup>e</sup> Dimension, ces propriétés s'ajoutent aux autres options d'interface, conservées pour des raisons de compatibilité. Comme dans les versions précédentes de 4<sup>e</sup> Dimension, les anciennes propriétés de plate-forme peuvent toujours être définies au niveau des Préférences de

la base, du formulaire ou de l'objet.



Seule l'option **Système** apparaît (en plus des anciennes options) dans les Préférences de la base. La nouvelle option **Impression** ne peut être définie qu'au niveau de chaque formulaire ou objet.

Le fonctionnement des commandes FIXER INTERFACE et Lire interface est inchangé — la commande Lire interface retourne -1 (Plate forme automatique) si l'option **Système** est définie pour la base dans les Préférences. Toutefois, leur utilisation est désormais déconseillée car elles ne seront pas maintenues dans les prochaines versions du programme. De manière générale, il est conseillé d'utiliser les nouvelles propriétés pour assurer la compatibilité des bases avec les prochaines versions de 4<sup>e</sup> Dimension.

### Contrôles de saisie natifs

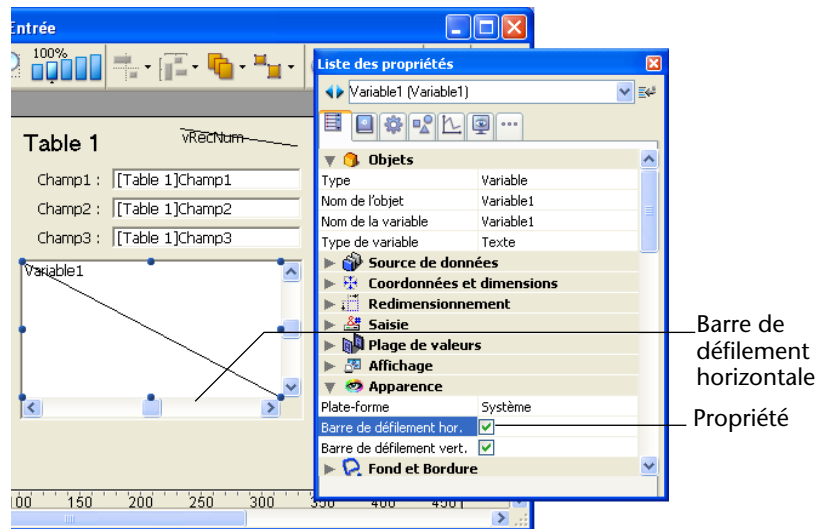
4<sup>e</sup> Dimension 2004 inclut des contrôles de saisie natifs pour les zones suivantes :

- champs
- variables
- list box

Une zone de texte avec contrôle natif signifie que la zone bénéficie de davantage d'options et d'une meilleure intégration à l'interface de la plate-forme courante. Les bases ouvertes avec 4<sup>e</sup> Dimension 2004 tirent automatiquement parti de ce nouveau mode de contrôle de saisie.

Les principaux avantages procurés par ce nouveau mode sont les suivants :

- **Couleurs de sélection système** : lorsque le texte de la zone est sélectionné (au premier plan ou à l'arrière-plan), la couleur de surlignage correspond à celle définie dans le système.
- **Barre de défilement horizontale** (type texte uniquement) : chaque zone de texte en contrôle natif peut bénéficier d'une barre de défilement horizontale. La barre de défilement horizontale est spécifiée dans le thème "Apparence" des propriétés de l'objet :



- **Glisser-déposer "système"** : les zones de texte en contrôle natif autorisent le glisser-déposer "système", c'est-à-dire le déplacement ou la copie de la sélection de texte d'une zone à une autre. Il peut être employé dans la même zone 4D, entre deux zones 4D, ou entre 4D et une autre application, par exemple WordPad. Ce principe ne fonctionne que lorsque la zone de départ et la zone d'arrivée sont en contrôle natif.



Lorsque le glisser-déposer système est utilisé, les mécanismes de gestion du glisser-déposer internes de 4<sup>e</sup> Dimension ne sont pas mis en oeuvre.

Par conséquent, les événements formulaire du type Sur glisser ne sont pas déclenchés et la zone de déposer n'est pas activée comme défini dans les Préférences. Si vous souhaitez "forcer" l'utilisation du glisser-déposer interne de 4<sup>e</sup> Dimension avec des zones en contrôle natif, appuyez sur la touche **Alt** (Windows) ou **Option** (MacOS) avant d'effectuer un glisser-déposer.

- **Détection et activation des URLs** : lorsqu'une zone texte en contrôle natif contient un URL (chaîne débutant par http, ftp, www ou mailto), celui-ci est automatiquement détecté ; sous Windows, il est affiché en bleu et souligné :



Si l'utilisateur effectue **Ctrl+clic** (Windows) ou **Commande+clic** (MacOS) dans la zone, l'URL est alors directement exécuté dans le navigateur Web par défaut.

---

*Note* Les adresses de messagerie doivent être écrites sous la forme `mailto:adresse`

---

### Correction orthographique

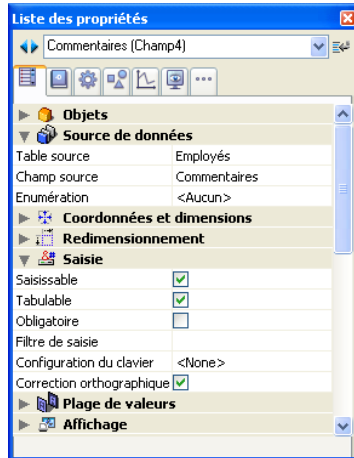
4<sup>e</sup> Dimension 2004 inclut un utilitaire de correction orthographique intégré, disponible en plusieurs langues. La vérification orthographique peut être effectuée pour les champs et variables de type Alpha et Texte — ainsi que pour les documents 4D Write.



## Activer la vérification orthographique

Pour activer la vérification de l'orthographe d'un champ ou d'une variable de type Alpha ou Texte, vous disposez de deux possibilités :

- cocher la propriété **Correction orthographique** (thème "Saisie") pour le champ ou la variable :

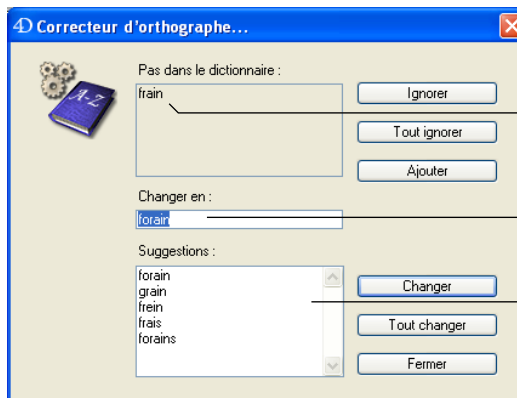


Dans ce cas, la vérification orthographique est effectuée automatiquement lors de la saisie, dès que l'objet perd le focus.

- exécuter la commande CORRECTION ORTHOGRAPHIQUE pour chaque objet à contrôler. Pour plus d'informations sur cette commande, reportez-vous au [paragraphe "CORRECTION ORTHOGRAPHIQUE"](#), page 425.

## Détection d'un mot inconnu

Quel que soit le mode d'activation du correcteur, en cas de détection d'un mot inconnu, la boîte de dialogue suivante apparaît :



Mot inconnu

Zone de correction

Autres suggestions (le cas échéant)

Plusieurs boutons sont disponibles :

- **Ignorer** : le mot inconnu est conservé tel quel.
- **Tout ignorer** : le mot inconnu est conservé tel quel et toutes les autres occurrences du mot dans la zone sont également conservées.
- **Ajouter** : le mot inconnu est ajouté au dictionnaire ; il ne sera plus détecté par le correcteur.
- **Changer** : le mot inconnu est remplacé par le mot présent dans la zone de correction.
- **Tout changer** : le mot inconnu et toutes les autres occurrences du mot dans la zone sont remplacés par le mot présent dans la zone de correction.
- **Fermer** : aucune correction n'est effectuée et la boîte de dialogue est refermée.

---

*Note* Lorsque vous avez cliqué sur un bouton, la vérification se poursuit jusqu'à la fin de la zone — à l'exception du bouton **Fermer**.

---

### Gestion des dictionnaires

Le correcteur de 4<sup>e</sup> Dimension 2004 peut utiliser quatre dictionnaires : français, anglais, allemand et espagnol.

Par défaut, 4<sup>e</sup> Dimension utilise le dictionnaire correspondant à la langue courante de l'application.

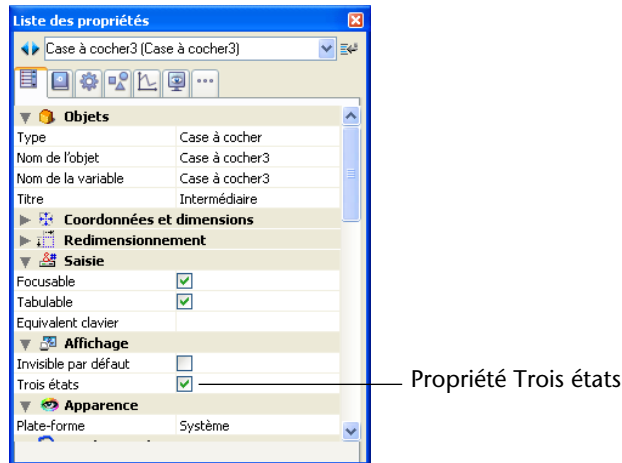
Vous pouvez toutefois forcer l'ouverture d'un dictionnaire différent à l'aide de la nouvelle commande **CHANGER DICTIONNAIRE**.

### Cases à cocher à trois états

Les objets de type case à cocher acceptent un troisième état dans 4<sup>e</sup> Dimension 2004. Ce troisième état représente un statut intermédiaire, généralement utilisé pour l'affichage. Il permet par exemple d'indiquer qu'une propriété est présente parmi une sélection d'objets mais pas dans tous les objets.



Pour qu'une case à cocher prenne en charge ce troisième état, vous devez lui attribuer la propriété **Trois états** dans la Liste des propriétés, thème "Affichage" :



Cette propriété n'est disponible que pour les cases à cocher standard associées à des variables numériques — les cases à cocher 3D et les cases à cocher de représentation des champs booléens sont exclues de ce principe (un champ booléen ne pouvant pas se trouver dans un état intermédiaire).

La variable associée à la case à cocher retourne la valeur 2 lorsque celle-ci se trouve dans le troisième état.

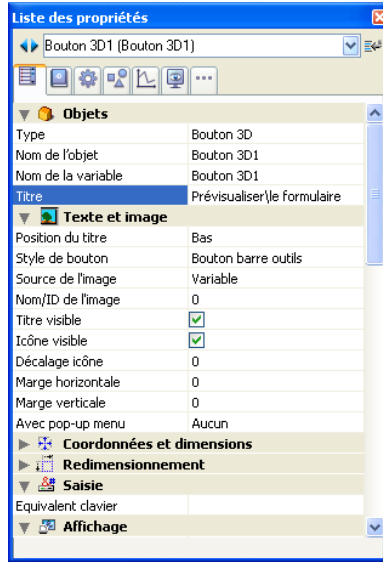
**Astuce :** En saisie, les cases à cocher à trois états affichent séquentiellement chaque état, dans l'ordre suivant : non coché / coché / intermédiaire / non coché, etc. L'état intermédiaire étant généralement inutile en saisie, il vous suffit, dans le code, de "forcer" la valeur de la variable à 0 lorsqu'elle prend la valeur 2 afin de passer directement de l'état coché à l'état non coché.

## Modification des boutons 3D

Les possibilités offertes par la famille des boutons 3D ont été étendues dans 4<sup>e</sup> Dimension 2004. Les modifications concernent les trois types de boutons 3D :

- Boutons 3D
- Boutons radio 3D
- Cases à cocher 3D

Ces nouveautés se traduisent notamment par la possibilité d'appliquer différents styles prédéfinis aux boutons 3D (bouton bevel, bouton poussoir, etc.) ou de leur associer des pop up menus. De nombreux types de boutons 3D peuvent être obtenus en combinant les nouvelles propriétés de la Liste des propriétés :



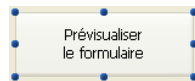
Nouvelles propriétés  
des boutons 3D

*Note de compatibilité* La plupart des nouvelles propriétés ne sont pas disponibles lorsque le Style de bouton **Décalage du fond** est sélectionné. Ce style, correspondant au fonctionnement précédent des boutons 3D, est sélectionné par défaut pour les boutons 3D des bases de données converties.

### Titre

Cette nouvelle propriété permet d'inscrire un libellé dans le bouton. La police et le style de ce libellé peuvent être définis dans le thème "Texte".

Vous pouvez forcer le passage à la ligne dans le libellé à l'aide du caractère \ (barre oblique inversée).



Pour insérer un \ dans le libellé, saisissez \\.

Par défaut, le libellé est placé au centre du bouton. Lorsque le bouton comporte également une icône, vous pouvez modifier l'emplacement relatif de ces deux éléments à l'aide de la propriété **Position titre/image**.

Il est possible de masquer le titre en désélectionnant l'option **Titre visible**. Dans ce cas, l'icône est automatiquement replacée au centre du bouton.

## Style de bouton

Cette propriété permet de définir l'apparence générale du bouton. Le style influe également sur la disponibilité de certaines options. Les styles suivants sont proposés :

- **Aucun** (valeur par défaut).



Un bouton 3D avec le style "Aucun" est semblable à un bouton invisible : son contour n'est pas affiché. Il bénéficie cependant des nouvelles options des boutons 3D.

Sous MacOS, il n'est pas possible d'afficher le triangle indiquant l'association à un pop up menu.

- **Décalage du fond**



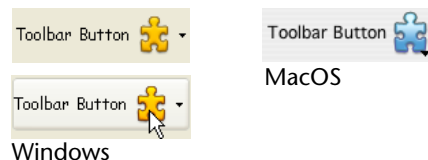
Ce style correspond aux boutons 3D disponibles dans les versions précédentes de 4<sup>e</sup> Dimension. Les boutons 3D des bases de données converties ont ce style par défaut. Lorsque le style "Décalage du fond" est sélectionné, aucune nouvelle option n'est disponible.

- **Bouton poussoir**



Un bouton 3D avec le style "Bouton poussoir" prend l'apparence d'un bouton système standard. Il bénéficie cependant des nouvelles options des boutons 3D, à l'exception de la propriété "Avec pop-up menu".

- **Bouton barre outils**

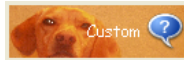


Ce style de bouton 3D est plus particulièrement destiné à une intégration dans une barre d'outils.

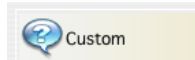
Sous Windows, son contour n'apparaît que lorsque la souris le survole. Lorsqu'il dispose de la propriété "Avec pop up menu", un triangle est affiché à droite et au centre du bouton.

Sous MacOS, le contour du bouton n'apparaît jamais. Lorsqu'il dispose de la propriété "Avec pop up menu", un triangle est affiché à droite et en bas du bouton.

### ■ Personnalisé



Windows



MacOS

Ce style de bouton 3D a pour particularité d'accepter une image d'arrière-plan personnalisée et permet de gérer divers paramètres supplémentaires (décalage de l'icône et marges). Pour plus d'informations, reportez-vous au [paragraphe "Boutons 3D personnalisés", page 163](#).

Ce style bénéficie des nouvelles options des boutons 3D, à l'exception de la propriété "Avec pop-up menu".

### ■ Rond



Windows

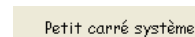


MacOS

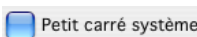
Sous MacOS X, un bouton 3D avec le style "Rond" apparaît sous la forme d'un bouton système rond. Deux tailles fixes sont disponibles pour le rond, dépendant de la taille du bouton dans le formulaire. Ce style bénéficie des nouvelles options des boutons 3D, à l'exception de la propriété "Avec pop-up menu".

Sous Windows, ce style de bouton est identique au style "Aucun" (le fond rond n'est pas pris en charge).

### ■ Petit carré système



Windows



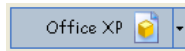
MacOS

Sous MacOS X, un bouton 3D avec le style "Petit carré système" apparaît sous la forme d'un carré système. Ce style bénéficie des nouvelles options des boutons 3D, à l'exception de la propriété "Avec pop-up

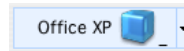
menu”.

Sous Windows, ce style de bouton est identique au style “Aucun” (le fond carré n’est pas pris en charge).

#### ■ Office XP



Windows



MacOS

Un bouton 3D avec le style “Office XP” a les caractéristiques suivantes :

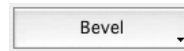
- les couleurs de son contour et de son fond sont basées sur les couleurs système.
- sous Windows, en utilisation, son contour n’apparaît que lorsque la souris le survole.

Ce style bénéficie des nouvelles options des boutons 3D.

#### ■ Bevel



Windows



MacOS

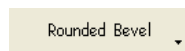


Windows

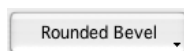
Sous MacOS, un bouton “Bevel” prend l’apparence d’un bouton système standard. Il bénéficie des nouvelles options des boutons 3D, y compris la propriété “Avec pop-up menu”.

Sous Windows, ce style de bouton a un fonctionnement semblable au style “Bouton barre outils”, à la seule différence que le triangle indiquant la présence d’un pop up menu associé est situé en bas à droite du bouton.

#### ■ Bevel arrondi



Windows



MacOS



Windows

Sous MacOS, un bouton “Bevel arrondi” est semblable à un bouton “Bevel”, à la différence près que ses contours sont arrondis.

Sous Windows, ce style de bouton est identique au style “Bevel”.

### Associer une icône au bouton 3D

Il est possible d'associer une icône à tous les styles de boutons 3D (à l'exception du style "Décalage du fond").



La gestion des icônes associées s'effectue via plusieurs propriétés :

#### ■ Source de l'image

Comme pour les boutons image, l'icône d'un bouton 3D peut provenir de trois sources différentes : **Variable**, **Bibliothèque d'images** ou **Fichiers de ressources**. Une fois la source désignée, vous pouvez indiquer le nom ou le numéro de l'image dans la propriété "Nom/ID de l'image".

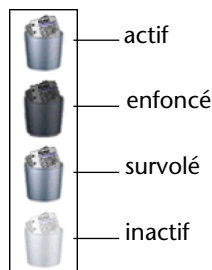
#### ■ Nom/ID de l'image

Une fois la provenance de l'image définie, saisissez dans cette zone le nom (si l'image est une variable ou provient de la bibliothèque d'images) ou le numéro (si l'image provient de la bibliothèque d'images ou d'un fichier de ressources) de l'image.

Pour ne pas associer d'image au bouton 3D, passez 0 dans cette propriété.

#### ■ Découpage de l'image

Quelle que soit la provenance de l'image, elle doit obligatoirement contenir quatre zones verticales distinctes, qui seront utilisées par 4<sup>e</sup> Dimension pour représenter les quatre états standard du bouton : **actif**, **enfoncé**, **survolé** et **désactivé**. Voici par exemple une image utilisée pour les formulaires 4<sup>e</sup> Dimension :



---

*Note* Le découpage de l'image est paramétré dans la liste des propriétés. Il n'est donc pas nécessaire de "découper" préalablement les images dans la bibliothèque d'images.

---



### ■ Position de l'image

Par défaut, l'image est placée au centre du bouton. Lorsque le bouton comporte également un libellé, vous pouvez modifier l'emplacement relatif de ces deux éléments à l'aide de la propriété **Position titre/image**.

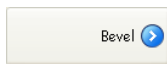
Il est possible de masquer l'icône en désélectionnant l'option  **Icône visible**. Dans ce cas, le texte est automatiquement remplacé au centre du bouton.

### Position titre/image

Cette propriété permet de modifier l'emplacement relatif du titre du bouton par rapport à l'icône associée. Cette propriété n'a pas d'effet lorsque le bouton contient uniquement un titre (pas d'image associée) ou une image (pas de titre).

Par défaut, lorsqu'un bouton 3D contient un titre et une image, le texte est placé au-dessous de l'image.

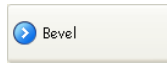
Voici le résultat des différentes options :



■ **Gauche** : Le texte est placé à gauche de l'icône. Le contenu du bouton est cadré à droite.



■ **Haut** : Le texte est placé au-dessus de l'icône. Le contenu du bouton est centré.



■ **Droite** : Le texte est placé à droite de l'icône. Le contenu du bouton est cadré à gauche.



■ **Bas** : Le texte est placé au-dessous de l'icône. Le contenu du bouton est centré.



■ **Centre** : Le texte et l'icône sont centrés verticalement et horizontalement dans le bouton. Ce paramétrage convient par exemple pour du texte inclus dans une icône.

### Avec pop-up menu

Cette nouvelle propriété permet d'afficher dans le bouton 3D un symbole en forme de triangle indiquant qu'un pop up menu lui est associé :



— Symbole d'association à un pop up menu

L'apparence et l'emplacement de ce symbole dépendent du style de bouton et de la plate-forme courante.

Les styles de boutons 3D acceptant la propriété “Avec pop up menu” sont les suivants :

- Aucun
- Bouton barre outils
- Bevel
- Bevel arrondi
- Office XP
- **Séparé et Lié**

Pour associer un symbole de pop up menu au bouton 3D, vous disposez de deux options d’affichage : **Séparé** et **Lié** :



---

*Note* La disponibilité effective d’un mode “Séparé” dépend du style de bouton et de la plate-forme.

---

En principe, chaque option permet de préciser la relation entre le bouton et le pop up menu associé :

- lorsque le pop up menu est *séparé*, un clic sur la partie gauche du bouton exécute directement l’action courante du bouton ; cette action peut être modifiée via le pop up menu accessible dans la partie droite du bouton.
- lorsque le pop up menu est *lié*, un simple clic sur le bouton ne déclenche aucune autre action que l’affichage du pop up menu. Seule la sélection de l’action dans le pop up menu provoque son déclenchement.

Ces options influent également sur la gestion des événements formulaire du bouton (pour plus d’informations sur ce point, reportez-vous au [paragraphe “Événements formulaire liés aux boutons 3D”](#), page 303).

- **Gestion du pop up menu**

Il est important de noter que la propriété “Avec pop up menu” gère uniquement l’aspect graphique du bouton. L’affichage du pop up menu et de ses valeurs doivent être entièrement gérées par le développeur, notamment à l’aide des nouveaux événements formulaires (cf. ci-dessus) et de la commande [Pop up menu](#).

## Boutons 3D personnalisés

Lorsque le style de bouton 3D “Personnalisé” est sélectionné, plusieurs propriétés supplémentaires sont disponibles : **Source du fond**, **Nom/ID du fond**, **Décalage icône**, **Marge horizontale** et **Marge verticale**.

- **Source du fond** : cette propriété vous permet de désigner l’image qui sera dessinée en arrière-plan du bouton. Comme pour l’icône, vous pouvez indiquer que l’image provient d’une variable, de la bibliothèque d’images ou d’un fichiers de ressources.
- **Nom/ID du fond** : une fois la source désignée, vous pouvez indiquer le nom ou le numéro de l’image dans cette zone.

Comme pour les icônes, les images d’arrière-plan doivent contenir quatre zones verticales distinctes, qui seront utilisées par 4<sup>e</sup> Dimension pour représenter les quatre états standard du bouton : **actif**, **enfoncé**, **survolé** et **désactivé**. A noter que l’effet produit lors du clic peut également être géré via la propriété **Décalage icône**.

- **Décalage icône**  
Cette propriété permet de définir une valeur de décalage (en pixels) personnalisée qui sera utilisée en cas de clic sur le bouton : le titre du bouton sera décalé vers la droite et vers le bas du nombre de pixels saisi. Cette fonction permet d’appliquer un effet 3D personnalisé lors du clic sur le bouton.
- **Marge horizontale / Marge verticale**  
Ces propriétés vous permettent de définir la taille (en pixels) des marges internes du bouton. Ces marges délimitent les zones dans lesquelles le titre et l’icône du bouton 3D ne doivent pas empiéter. Ce paramétrage est utile par exemple lorsque l’image d’arrière-plan contient des bordures :



Bouton 3D personnalisé sans marge



Bouton 3D avec marge 13 pixels

**Autres modifications** Diverses modifications ont été apportées aux objets de formulaires dans 4<sup>e</sup> Dimension 2004. Elles sont listées dans ce paragraphe.

## Type booléen pour les variables d’objets

Il est désormais possible d’assigner le type booléen (à l’aide de la commande C\_BOOLEAN) aux variables associées aux boutons radio, cases à cocher et boutons standard. Dans les versions précédentes de 4<sup>e</sup> Dimension, ces variables devaient être de type numérique.

Lorsqu'une variable de type booléen est utilisée, elle vaut Vrai lorsque l'objet est coché ou enfoncé et Faux sinon.

### Calcul des références incluses

Dans les versions précédentes de 4<sup>e</sup> Dimension, les références incluses dans les zones de texte statique (par exemple "<var\_genre> [Clients]Nom") étaient calculées une seule fois, au chargement du formulaire.

Désormais, ces références sont recalculées dès qu'une référence associée change — fonctionnement identique à celui d'une variable.

### Nouveau mode de fonctionnement des boutons radio

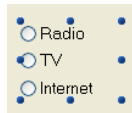
Dans les versions précédentes de 4<sup>e</sup> Dimension, le fonctionnement coordonné d'un groupe de boutons radio était obtenu en donnant une première lettre identique aux variables associées aux boutons (par exemple *m\_bouton1*, *m\_bouton2*, *m\_bouton3*, etc.).

---

*Note* Le fonctionnement coordonné implique qu'un seul bouton peut être sélectionné à la fois parmi l'ensemble.

---

Ce principe pouvant s'avérer insatisfaisant dans certains cas, il a été remplacé dans 4<sup>e</sup> Dimension 2004 par le suivant : pour fonctionner de manière coordonnée, un ensemble de boutons radio doit simplement être **groupé** dans l'éditeur de formulaires. Pour cela, vous pouvez utiliser la commande **Grouper** dans le menu **Objets** ou le bouton correspondant de la barre d'outils du formulaire.



---

*Note* Il est désormais possible de sélectionner un objet appartenant à un groupe sans devoir dégroupier l'ensemble : pour cela, effectuez **Ctrl+clac** (Windows) ou **Commande+clac** (MacOS) sur l'objet (le groupe doit être sélectionné au préalable).

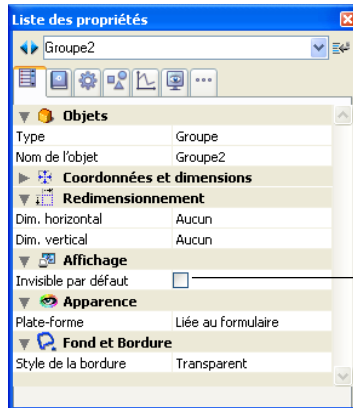
---

Pour des raisons de compatibilité, l'ancien mode est conservé par défaut dans les bases de données converties. Vous pouvez cependant forcer l'utilisation du nouveau mode en désélectionnant l'option **Boutons radio groupés par nom** située dans les Préférences (thème "Application", page Compatibilité). Cette option n'apparaît pas dans les bases de données créées avec 4<sup>e</sup> Dimension 2004.

Ce nouveau mode est valide pour les boutons radio, les boutons radio 3D et les boutons radio image.

## Nouvelle propriété Invisible par défaut

La Liste des propriétés fait apparaître la propriété **Invisible par défaut** dans le thème "Affichage" pour la plupart des objets :



Propriété Invisible par défaut

Cette propriété répond à un besoin de simplification du développement d'interfaces dynamiques. Dans ce cadre en effet, il est souvent nécessaire de masquer par programmation des objets lors de l'événement Sur chargement du formulaire puis d'en afficher certains par la suite.

La propriété **Invisible par défaut** permet d'inverser cette logique en rendant invisible par défaut certains objets. Le développeur peut ensuite programmer leur affichage à l'aide de la commande CHOIX VISIBLE en fonction du contexte.

*Note* Ne confondez pas cette propriété avec l'attribut **Invisible** des pop up menus, qui permet de ne pas dessiner l'objet tout en le laissant actif.

## Gestion des listes d'enregistrements

La gestion des listes d'enregistrements a été modifiée dans 4<sup>e</sup> Dimension 2004 de manière à simplifier et améliorer le fonctionnement des différents types de listes.

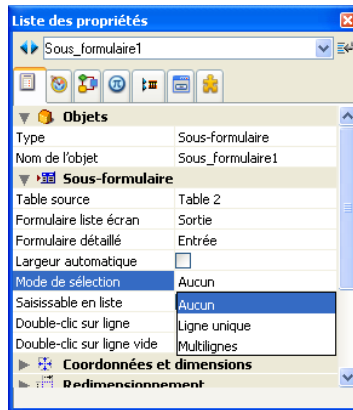
Dans les versions précédentes de 4<sup>e</sup> Dimension, les enregistrements pouvaient être affichés dans trois environnements, chacun ayant des propriétés spécifiques :

- les formulaires liste en mode Utilisation,
- les formulaires liste affichés via la commande MODIFIER SELECTION ou VISUALISER SELECTION,
- les sous-formulaires (aussi appelés formulaire inclus).

Dans 4<sup>e</sup> Dimension 2004, le fonctionnement de ces trois environnements a été harmonisé. Leur comportement est désormais quasiment identique, le maintien de la compatibilité avec les versions précédentes de 4<sup>e</sup> Dimension entraînant toutefois quelques différences.

**Modes de sélection** Les listes d'enregistrements acceptent désormais trois modes de sélection : *Aucun*, *Ligne unique* et *Multilignes*.

- Pour les sous-formulaires, le mode de sélection est une propriété applicable lorsque l'objet sous-formulaire est sélectionné :



- Pour les formulaires liste affichés via les commandes MODIFIER SELECTION et VISUALISER SELECTION, un nouveau paramètre permet de définir le mode de sélection (cf. [paragraphe "MODIFIER SELECTION, VISUALISER SELECTION"](#), page 358).
- Dans les formulaires liste en mode Utilisation, le mode *Multilignes* est automatiquement utilisé.

### Aucun

Dans ce mode, aucun enregistrement ne peut être sélectionné. L'ensemble système *UserSet* n'est pas défini. Un clic dans la liste n'a pas d'effet sauf si le mode "Saisie en liste" est actif.

Les touches de navigation provoquent uniquement le défilement de la liste, le nouvel événement formulaire **Sur nouvelle sélection** n'est pas généré (pour plus d'informations sur cet événement formulaire, reportez-vous au [paragraphe "Evénements formulaire liés aux enregistrements en liste"](#), page 306).

*Note* Dans les versions précédentes de 4<sup>e</sup> Dimension, ce mode était disponible pour les sous-formulaires via la propriété "Non modifiable".

---

### Ligne unique

Dans ce mode, un seul enregistrement à la fois peut être sélectionné. L'ensemble système *UserSet* n'est pas défini. Un clic sur un enregistrement provoque sa sélection et il devient l'enregistrement courant. La combinaison **Ctrl+clic** (Windows) ou **Commande+clic** (MacOS) sur un enregistrement inverse son état sélectionné / désélectionné.

Les touches fléchées **Haut** et **Bas** sélectionnent l'enregistrement précédent/suivant dans la liste. Les autres touches de navigation font défiler la liste. Le nouvel événement formulaire **Sur nouvelle sélection** est généré à chaque changement d'enregistrement courant.

*Note* Dans les versions précédentes de 4<sup>e</sup> Dimension, ce mode était disponible pour les sous-formulaires via la propriété "Sélectionnable".

---

### Multilignes

Dans ce mode, plusieurs enregistrements peuvent être sélectionnés simultanément. L'ensemble système *UserSet* est disponible. Un clic sur un enregistrement provoque sa sélection mais ne modifie pas l'enregistrement courant. La combinaison **Ctrl+clic** (Windows) ou **Commande+clic** (MacOS) sur un enregistrement inverse son état sélectionné/désélectionné. Les combinaisons **Ctrl+clic** (Windows) ou **Commande+clic** (MacOS) et **Maj+clic** permettent de créer des sélections multiples.

Les touches fléchées **Haut** et **Bas** sélectionnent l'enregistrement précédent/suivant dans la liste. Les autres touches de navigation font défiler la liste. Le nouvel événement formulaire **Sur nouvelle sélection** est généré à chaque changement d'enregistrement sélectionné.

*Note* Dans les versions précédentes de 4<sup>e</sup> Dimension, ce mode était disponible pour les listes affichées via les commandes MODIFIER SELECTION et VISUALISER SELECTION uniquement.

---

La commande **MARQUER ENREGISTREMENTS** et la nouvelle commande **LIRE ENREGISTREMENTS MARQUES** permettent de gérer par programmation les enregistrements sélectionnés.

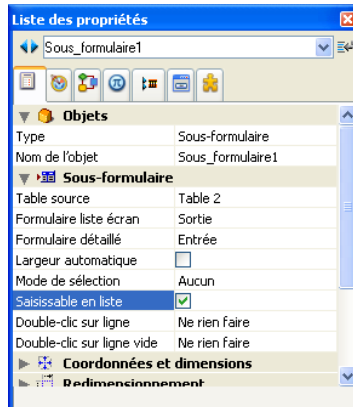
Après l'exécution de la méthode formulaire ou d'une méthode objet, 4<sup>e</sup> Dimension vérifie si la sélection courante a été modifiée. Si c'est le cas et si la commande **MARQUER ENREGISTREMENTS** n'a pas été appelée, 4<sup>e</sup> Dimension initialise la sélection d'enregistrements et génère l'événement formulaire **Sur nouvelle sélection**.

*Note* La commande **ALLER DANS SELECTION** permet désormais de désélectionner tous les enregistrements d'une liste.

### Saisie en liste

Le mode "Saisie en liste" est désormais disponible dans tous les types de listes d'enregistrements. Il n'était auparavant utilisable qu'en mode Utilisation.

- Pour les sous-formulaires, le mode Saisie en liste est une propriété applicable lorsque l'objet sous-formulaire est sélectionné :



- Pour les formulaires liste affichés via les commandes **MODIFIER SELECTION** et **VISUALISER SELECTION**, un nouveau paramètre permet d'activer ou de désactiver le mode Saisie en liste (cf. [paragraphe "MODIFIER SELECTION, VISUALISER SELECTION"](#), page 358).
- Dans les formulaires liste en mode Utilisation, le mode Saisie en liste est automatiquement activé.

En mode Saisie en liste, l'utilisateur peut modifier les données des enregistrements directement dans la liste, sans devoir passer par un formulaire en page. Pour cela, il suffit de cliquer sur le champ à modifier afin de le passer en édition. Par défaut, le mode Saisie en liste est activé pour toutes les listes.



Au moment où l'utilisateur clique sur une ligne pour la passer en mode édition, l'enregistrement correspondant est chargé. La méthode formulaire et les méthodes objets sont appelées et le nouvel événement formulaire **Sur chargement ligne** est généré (cf. [paragraphe "Événements formulaire liés aux enregistrements en liste", page 306](#)).

## Affichage des listes

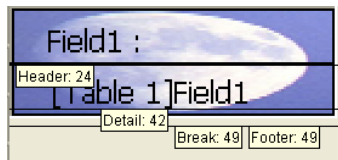
Les mécanismes d'affichage des enregistrements en liste ont été optimisés dans 4<sup>e</sup> Dimension 2004. Ces modifications concernent :

- l'affichage des objets situés dans la zone de corps des formulaires liste,
- l'affichage des lignes vides,
- la visibilité des objets en fonction de leur sélection,
- le déclenchement des événements formulaires,
- les objets actifs acceptés dans les différentes zones d'un formulaire en liste.

## Objets situés dans le corps des formulaires

Les objets dont la partie supérieure débute dans la zone d'en-tête et la partie inférieure se termine dans la zone de rupture sont désormais dessinés en une seule fois, par extension de la hauteur de l'objet. Auparavant, la partie située dans la zone de corps était répliquée.

Cette modification est perceptible uniquement pour les images, comme illustré ci-dessous :



Editeur de formulaires



4<sup>e</sup> Dimension 2003

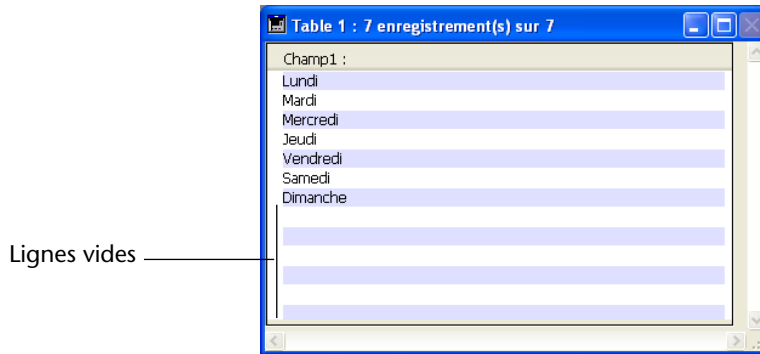


4<sup>e</sup> Dimension 2004

Ce mécanisme tient compte des propriétés d'affichage de l'image (tronquée, mosaïque, etc.).

### Lignes vides

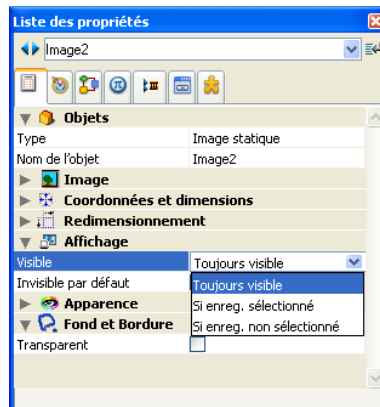
En mode liste, 4<sup>e</sup> Dimension affiche désormais si nécessaire des lignes vides au-dessous des derniers enregistrements affichés, de manière à remplir entièrement la fenêtre :



Lors de l'affichage de la fenêtre, l'événement formulaire **Sur affichage corps** est désormais également généré pour chaque ligne vide de la liste. Dans ce cas, il n'y a pas d'enregistrement courant : Numero enregistrement retourne -1 et Numero dans selection ne retourne pas de valeur significative. La nouvelle commande **Numero de ligne affichée** permet dans ce cas de connaître le numéro de la ligne en cours d'affichage.

### Visibilité des objets

La nouvelle propriété d'affichage **Visible** est disponible pour tous les objets dans l'éditeur de formulaires lorsque le type du formulaire est "liste" :

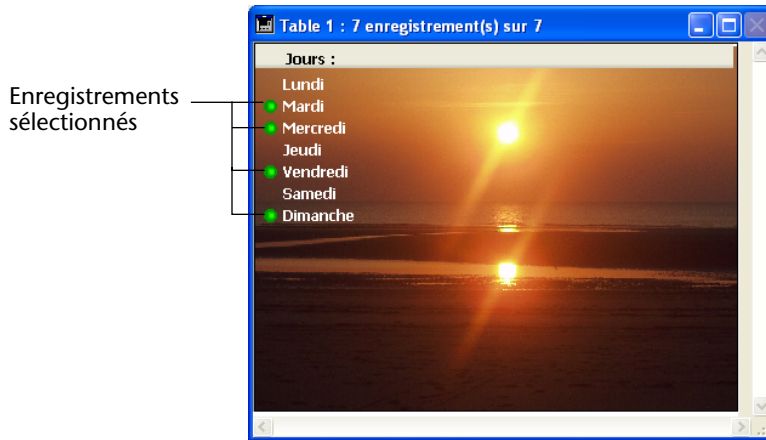


Cette propriété accepte trois valeurs :

- Toujours visible
- Si enreg. sélectionné

- Si enreg. non sélectionné

Cette propriété est utilisée uniquement lors du dessin des objets situés dans la zone de corps d'un formulaire liste. Elle indique à 4<sup>e</sup> Dimension de dessiner ou non l'objet en fonction de l'état sélectionné/non sélectionné de l'enregistrement en cours de traitement. Elle permet notamment de représenter la sélection d'enregistrements à l'aide d'attributs visuels autres que les couleurs :



4<sup>e</sup> Dimension ne tient pas compte de cette propriété si l'objet a été caché à l'aide de la commande CHOIX VISIBLE ; dans ce cas, l'objet reste invisible quel que soit l'état de sélection de l'enregistrement.

En cas d'affichage de lignes vides après les derniers enregistrements (cf. [paragraphe "Lignes vides", page 170](#)), 4<sup>e</sup> Dimension dessine dans chaque ligne vide les objets ayant la propriété **Toujours visible** ou **Si enreg. non sélectionné**.

### Séquence d'appel des événements formulaires

L'enchaînement d'appels des méthodes et des événements formulaires dans les sous-formulaires a été aligné sur celui des formulaires liste affichés via MODIFIER SELECTION / VISUALISER SELECTION dans 4<sup>e</sup> Dimension 2004. Voici les différences avec les versions précédentes :

- Appels dans les sous-formulaires avec 4<sup>e</sup> Dimension < 2004

Pour chaque enregistrement :

- Formule-fichier (ancien mécanisme)

- Méthode formulaire avec événement *Sur affichage corps*

- **Appels dans les sous-formulaires avec 4<sup>e</sup> Dimension 2004 (= appels dans les formulaires liste affichés avec MODIFIER SELECTION/VISUALISER SELECTION) :**

Pour chaque objet de la zone d'en-tête :

    Méthode objet avec événement *Sur entête*

Méthode formulaire avec événement *Sur entête*

Pour chaque enregistrement :

    Pour chaque objet de la zone de corps :

        Méthode objet avec événement *Sur affichage corps*

        Méthode formulaire avec événement *Sur affichage corps*

A noter que les anciennes formules-fichiers (type V5.x.x) ne sont plus appelées pour les sous-formulaires.

### **Objets actifs**

La prise en charge des objets actifs dans les zones d'en-tête, de rupture et de pied de page des formulaires en liste a été étendue :

- Les formulaires liste affichés via MODIFIER SELECTION / VISUALISER SELECTION et les sous-formulaires acceptent tous les types d'objets actifs dans les zones d'en-tête et de rupture.
- Les formulaires liste affichés via MODIFIER SELECTION / VISUALISER SELECTION acceptent tous les types d'objets actifs dans les zones de pied de page.

### **Options des sous-formulaires**

Les sous-formulaires disposent d'attributs supplémentaires permettant un meilleur contrôle sur leur apparence et de leur fonctionnement.

### **Barre de défilement horizontale**

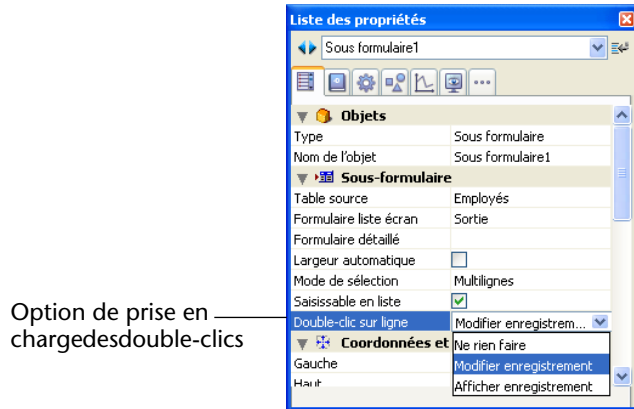
Une nouvelle option dans la liste des propriétés (thème "Apparence") permet d'associer une **barre de défilement horizontale** à un sous-formulaire. Lorsque vous cochez cette option, une barre de défilement est ajoutée sous le sous-formulaire, à l'extérieur de la zone d'origine.

### **Action en cas de double-clic**

Vous pouvez désormais paramétrer le fonctionnement des sous-formulaires en réponse à un double-clic de l'utilisateur.

Dans les bases de données créées avec une ancienne version de 4<sup>e</sup> Dimension, vous pouvez en outre paramétrer la réponse en cas de double-clic sur une ligne vide.

Ces paramètres sont accessibles dans le thème “Sous-formulaire” de la Liste des propriétés :



- **Double-clic sur ligne** : action à effectuer en cas de double-clic sur un enregistrement du sous-formulaire. Les options proposées sont :
  - **Ne rien faire** : ignorer le double-clic.
  - **Modifier enregistrement** : passer le sous-enregistrement en mode édition. La modification sera effectuée directement dans la liste si l’option “Saisissable en liste” est cochée. Sinon, elle sera effectuée en mode page, dans le formulaire détaillé associé au sous-formulaire.
  - **Afficher enregistrement** : afficher les données de l’enregistrement en mode page dans le formulaire détaillé associé au sous-formulaire (lecture seulement).
- **Double-clic sur ligne vide (formulaires convertis uniquement)** : dans les versions précédentes de 4<sup>e</sup> Dimension, l’ajout de sous-enregistrements pouvait être effectué via un double-clic sur une ligne de sous-formulaire vide. Pour des raisons de compatibilité, ce mécanisme est maintenu pour les formulaires créés avec une ancienne version de 4<sup>e</sup> Dimension — il est toutefois conseillé d’utiliser désormais des actions standard pour gérer les ajouts (cf. [paragraphe “Actions standard”, page 174](#)). Vous pouvez paramétrer l’action à effectuer en cas de double-clic sur une ligne vide du sous-formulaire :
  - **Ne rien faire** : ignorer le double-clic.
  - **Ajouter enregistrement** : créer un nouveau sous-enregistrement et le passer en édition. La création sera effectuée directement dans la liste si l’option “Saisissable en liste” est cochée. Sinon, elle sera effectuée en mode page, dans le formulaire détaillé associé au sous-formulaire.

**Redimensionnement** Les mécanismes de redimensionnement des sous-formulaires ont été modifiés afin de mieux prendre en compte les options définies dans la Liste des propriétés.

**Focus** L'objet sous-formulaire peut désormais disposer de la propriété **Focusable**. Lorsqu'un sous-formulaire a le focus, l'utilisateur peut le contrôler à l'aide des touches de navigation, utiliser la commande de menu **Tout sélectionner** (si la sélection est multilignes), etc.

Lorsqu'un sous-formulaire reçoit ou perd le focus, la méthode formulaire du formulaire hôte est appelée avec les événements Sur gain focus ou Sur perte focus. Dans ce cas, la commande **Objet focus** (nouveau nom de la commande Dernier objet) retourne un pointeur vers la table ou la sous-table du sous-formulaire.

Les sous-formulaires n'affichent plus de triangle clignotant pour indiquer qu'ils ont pris le focus. Cette information est désormais représentée par le changement de couleur du sous-enregistrement sélectionné. Il est également possible d'utiliser un indicateur visuel personnalisé (à l'aide de l'événement Sur gain focus par exemple).

**Actions standard** Dans 4<sup>e</sup> Dimension 2004, le fonctionnement des actions standard *Ajouter sous-enregistrement* et *Modifier sous-enregistrement* a été modifié.

- Outre les sous-formulaires, ces actions fonctionnent désormais avec les formulaires en liste affichés via MODIFIER SELECTION / VISUALISER SELECTION et en mode Utilisation.
- L'ajout ou la modification d'un enregistrement s'effectue en liste ou en page en fonction du statut de l'option "Saisissable en liste" du formulaire. Cette option est définie soit dans les propriétés des sous-formulaires, soit via le nouveau paramètre *saisieListe* des commandes MODIFIER SELECTION / VISUALISER SELECTION, ou encore dans le menu **Enregistrements** (mode Utilisation) :
  - si la saisie en liste est activée, l'ajout ou la modification s'effectue en liste,
  - si la saisie en liste est désactivée, l'ajout ou la modification s'effectue en mode page.
- Ces actions sont disponibles pour les boutons et les commandes de menus.

# Bibliothèques d'objets

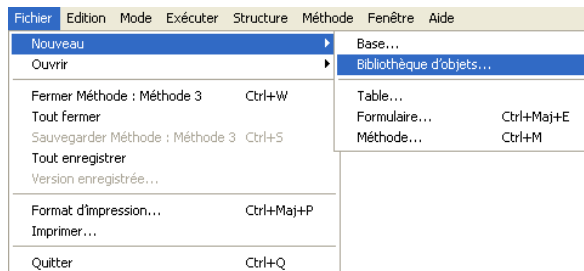
4<sup>e</sup> Dimension 2004 permet de créer et d'utiliser des *bibliothèques d'objets*.

Une bibliothèque d'objets est un fichier externe créé par 4<sup>e</sup> Dimension. Elle permet de stocker tous les types d'objets utilisables dans les formulaires : boutons, textes, images, listes hiérarchiques, etc. Les objets sont stockés avec toutes leurs propriétés, y compris leurs méthodes objet. Les bibliothèques sont constituées et utilisées par simple glisser-déposer ou copier-coller. Ce sont en quelque sorte des presse-papiers permanents.

A l'aide des bibliothèques, vous pouvez constituer des fonds d'objets de formulaires regroupés par familles graphiques, par fonctionnalités, etc. Ces bibliothèques étant stockées sous forme de fichiers externes, leur utilisation parmi différentes bases est d'une grande simplicité.

## Créer une bibliothèque d'objets

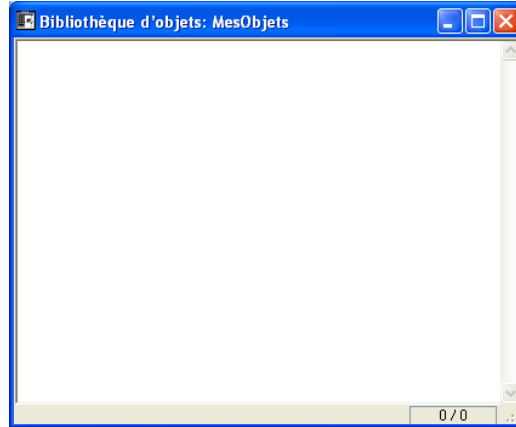
Pour créer une bibliothèque d'objets, sélectionnez la commande **Nouveau>Bibliothèque d'objets...** dans le menu **Fichier** de 4<sup>e</sup> Dimension.



Une boîte de dialogue standard d'enregistrement de fichiers apparaît, vous permettant de choisir le nom et l'emplacement de la bibliothèque d'objets. Sous Windows, l'extension des bibliothèques est **".4il"**.

*Note* Cette extension est identique à celles des librairies d'objets de 4D Insider. En effet, ces deux types de fichiers sont semblables.

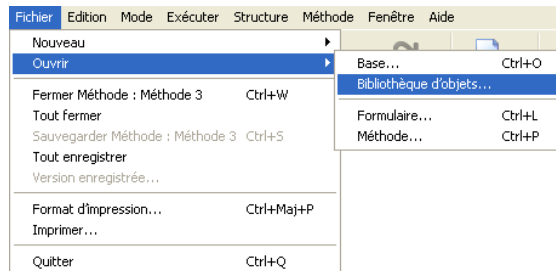
Lorsque vous validez la boîte de dialogue, 4<sup>e</sup> Dimension crée sur disque la nouvelle bibliothèque d'objets et affiche sa fenêtre, vide par défaut.



Vous pouvez créer autant de bibliothèques que vous voulez par base. Une bibliothèque créée et construite sous MacOS peut être utilisée sous Windows et inversement.

### Ouvrir une bibliothèque d'objets

Pour ouvrir une bibliothèque d'objets existante, sélectionnez la commande **Ouvrir>Bibliothèque d'objets...** dans le menu **Fichier** de 4<sup>e</sup> Dimension.



Une boîte de dialogue standard d'ouverture de fichiers apparaît, vous permettant de désigner la bibliothèque d'objets à ouvrir.

Une même bibliothèque d'objets ne peut être ouverte que par une seule base à la fois. En revanche, il est possible d'ouvrir plusieurs bibliothèques différentes dans la même base.

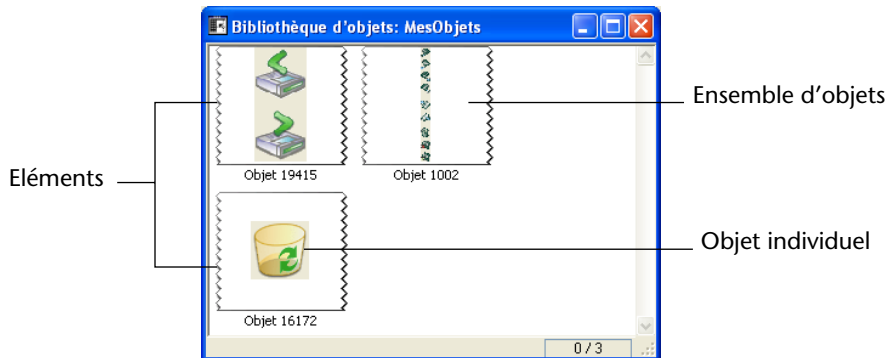
### Construire une bibliothèque d'objets

Les objets sont placés dans une bibliothèque d'objets via un glisser-déposer ou un couper-copier-coller. Ils peuvent provenir soit d'un formulaire soit d'une autre bibliothèque d'objets. Aucun lien n'est



conservé avec l'objet d'origine : si celui-ci est modifié, la modification ne sera pas reportée dans l'objet copié.

Vous pouvez placer dans la bibliothèque des objets individuels ou des ensembles d'objets. Chaque objet ou ensemble d'objets est regroupé en un seul élément :

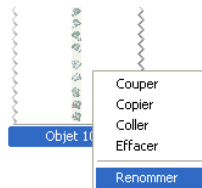


Une bibliothèque d'objets peut contenir jusqu'à 32 000 éléments.

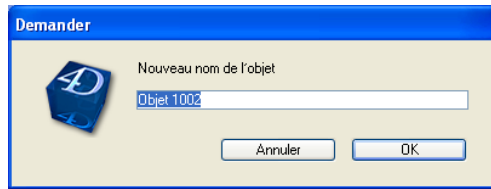
Les objets sont copiés avec toutes leurs propriétés, graphiques et fonctionnelles. Elles sont intégralement conservées lorsque l'élément est recopié dans un formulaire ou une autre bibliothèque.

Chaque nouvel élément est nommé "Objet" suivi d'un numéro aléatoire, par exemple *Objet 2012*. Vous pouvez renommer cet élément comme vous le souhaitez. Pour cela, vous pouvez :

- double-cliquer sur l'élément à renommer
- cliquer avec le **bouton droit** de la souris (Windows) ou effectuer **Control+clic** (MacOS) sur l'élément à renommer puis choisir la commande **Renommer...** dans le menu contextuel :



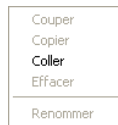
Une boîte de dialogue vous permet de renommer l'élément :



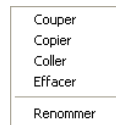
Plusieurs éléments peuvent avoir le même nom.

### Menu contextuel

Chaque bibliothèque d'objets dispose d'un menu contextuel permettant d'accéder aux principales commandes de gestion. Le contenu de ce menu dépend de l'emplacement du clic :



Clic en-dehors de  
tout élément



Clic sur un élément

---

*Note* Il n'est pas possible de sélectionner simultanément plusieurs éléments.

---

- **Couper / Copier** : ces commandes permettent de placer l'élément sélectionné de la bibliothèque en mémoire. La commande **Couper** supprime l'élément de la bibliothèque. Dans les deux cas, l'élément peut ensuite être collé dans un formulaire ou dans une autre bibliothèque d'objets.
- **Coller** : cette commande n'est active que si un objet compatible avec la bibliothèque se trouve en mémoire. Elle permet d'insérer dans la bibliothèque l'objet en mémoire.

---

*Note* Ces commandes sont également accessibles via le menu **Edition** et les raccourcis-clavier standard.

---

- **Effacer** : cette commande supprime l'élément sélectionné de la bibliothèque. Vous pouvez également utiliser la touche **Suppr.** ou **Ret. Arr.** du clavier.
- **Renommer...** : cette commande affiche la boîte de dialogue permettant de renommer l'élément (cf. paragraphe précédent).

## Objets dépendants

Le copier-coller ou le glisser-déposer de certains objets dans la bibliothèque entraîne la copie des objets dépendants. Par exemple, la copie d'un bouton entraînera obligatoirement la copie de la méthode objet qui lui est éventuellement attachée. Ces objets dépendants ne peuvent, quant à eux, être directement copiés ou glissés-déposés.

Voici la liste des objets dépendants qui seront collés dans la bibliothèque en même temps que l'objet principal qui les utilise (le cas échéant) :

- Enumérations
- Feuilles de style
- Formats/Filtres
- Images
- Messages d'aide (liés à des champs)
- Ressources STR#
- Méthodes objet

Dans le cas d'une ressource STR#, c'est toute la STR# qui est copiée, pas seulement la chaîne associée à l'objet.

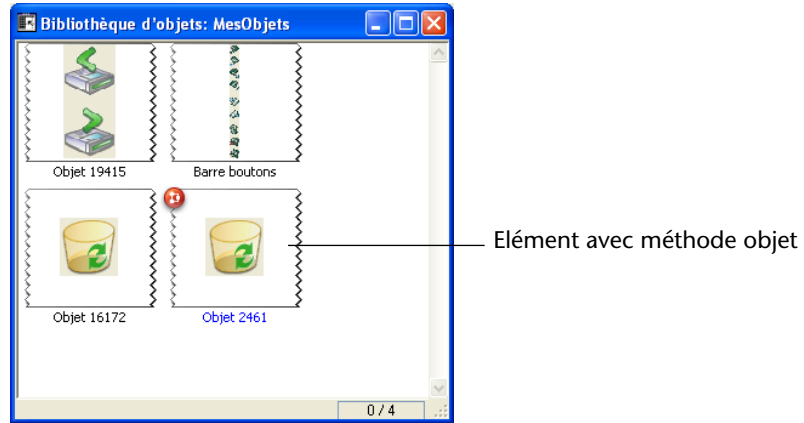
## Affichage des bibliothèques d'objets

Les bibliothèques d'objets sont accessibles à partir du mode Structure de 4<sup>e</sup> Dimension et, si le développeur le permet, dans l'éditeur de formulaires utilisateurs (cf. [paragraphe "Accès aux bibliothèques d'objets"](#), page 34).

Tous les éléments de la bibliothèque sont visibles en mode Structure. En revanche, seuls les éléments n'ayant pas de méthode objet associée sont visibles dans l'éditeur de formulaires utilisateurs.

Les éléments ayant au moins une méthode objet associée sont affichés avec un badge (le même que celui qui est affiché sur les objets ayant

une méthode objet dans un formulaire, cf. [paragraphe "Utilisation des badges"](#), page 125) et leur nom apparaît en bleu :



Ce principe permet de différencier facilement les éléments visibles par le développeur en mode Structure de ceux visibles par l'utilisateur en mode Utilisation.

### Fonctionnement en client/serveur

Afin que tous les postes clients puissent accéder à une bibliothèque d'objets, le fichier de la bibliothèque doit être placé dans un dossier nommé **Extras**, placé au même niveau que la structure de la base ouverte par 4D Server.

Lors de la connexion d'un 4D Client, le dossier Extras est rapatrié sur le poste client au même niveau que le dossier Plugins, dans le dossier 4D actif du système.

Ce principe fonctionne quelle que soit la plate-forme du serveur et du client.

# Générateur d'applications

4<sup>e</sup> Dimension 2004 propose un nouveau générateur d'applications plus performant, répondant à divers besoins notamment en matière de déploiement d'applications client/serveur.

Le nouveau générateur d'applications permet en particulier de :

- construire via un projet XML différentes applications à partir de la même base de données compilée,
- construire des applications client/serveur homogènes,
- générer des applications client/serveur avec mise à jour automatique des parties clientes.

## Nouveaux noms pour les applications de déploiement

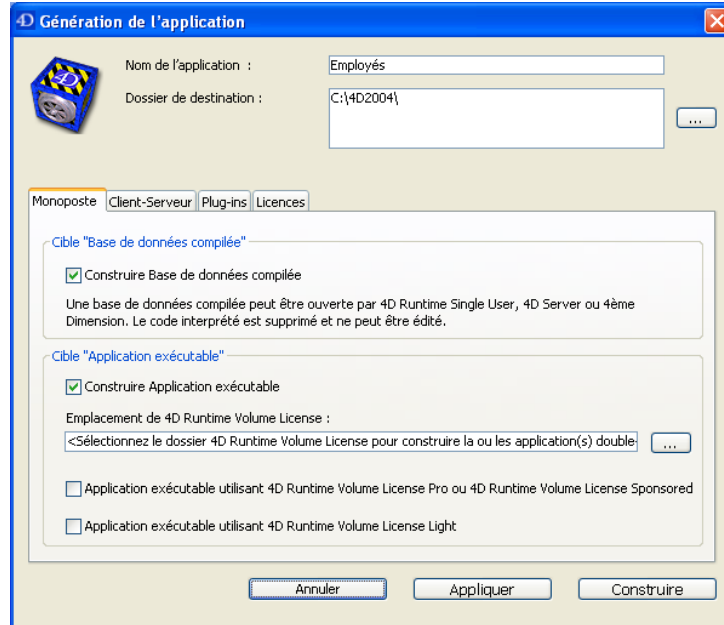
Dans un but de clarification et d'harmonisation de l'offre de déploiement en version monoposte, les applications de la gamme 4<sup>e</sup> Dimension suivantes ont été renommées (leur usage est inchangé) :

- **4D Runtime** devient **4D Runtime Interpreted**  
4D Runtime Interpreted permet d'exécuter une base de données 4D en interprété (cette application est fournie gratuitement et est librement redistribuable).
- **4D Runtime Classic** devient **4D Runtime Single User**  
4D Runtime Single User permet d'exécuter une base de données 4D compilée.
- **4D Engine** devient **4D Runtime Volume License**  
4D Runtime Volume License s'intègre à une base de données 4D compilée afin de générer une application exécutable autonome.  
A l'instar des versions 2003.x de 4<sup>e</sup> Dimension, 4D Runtime Volume License se décline en 4D Runtime Volume License Pro, 4D Runtime Volume License Sponsored et 4D Runtime Volume License Light.

Pour plus d'informations sur les conditions d'obtention et d'utilisation de ces applications, veuillez contacter 4D ou vous reporter au site <http://www.4d.fr>.

## Boîte de dialogue et projet de génération d'application

La nouvelle boîte de dialogue du générateur d'applications comporte désormais plusieurs onglets :



- La page **Monoposte** permet de paramétrer les versions monopostes de l'application (comme dans la version précédente du générateur).
- La page **Client-Serveur** permet de paramétrer la version client/serveur de l'application.
- La page **Plug-ins** permet de désigner les plug-ins à intégrer ou non à l'application.
- La page **Licences** permet de désigner les numéros de série à intégrer.

Il est possible de paramétrer simultanément différentes "cibles" (monopostes et client/serveur) pour l'application, chacune devant être explicitement sélectionnée via une case à cocher. Le bouton **Construire** génère les applications correspondant à toutes les cibles sélectionnées.

Chaque paramètre de cette fenêtre est stocké sous forme de clé XML dans un fichier de projet d'application nommé "BuildApp.XML" et placé dans le sous-dossier BuildApp du dossier Preferences de la base.

Les paramètres par défaut sont utilisés lors de la première utilisation de la boîte de dialogue. Le contenu de ce fichier de projet est mis à jour si nécessaire lorsque vous cliquez sur le bouton **Construire** ou **Appliquer**.

Le bouton **Appliquer** permet d'enregistrer les paramètres définis sans déclencher la construction.

Côté client, le fichier XML est stocké dans le dossier *4D Extensions* de l'application cliente.

Il est possible de définir plusieurs autres projets XML pour la même base de données et de les exploiter via la nouvelle [commande GENERER APPLICATION](#), page 423.

A noter que les clés XML proposent des options supplémentaires par rapport à celles affichées dans la boîte de dialogue du générateur d'applications, notamment :

- la définition de l'adresse IP ou du numéro de port TCP du serveur,
- la définition d'un intervalle de numéros de versions compatibles entre l'application cliente et l'application serveur (par exemple, les applications clientes 1.1 à 1.3 peuvent se connecter à l'application serveur 1.3).

La description de ces clés fait l'objet d'un manuel séparé, *4D Clés XML*.

## Fichier d'historique

Au moment de la première construction d'une application, 4<sup>e</sup> Dimension génère un fichier d'historique au format XML. Un fichier d'historique est généré pour chaque projet d'application.

Le fichier d'historique est nommé "NomProjet.log" (*NomProjet* représente le nom du projet d'application, c'est-à-dire BuildApp.xml par défaut) et est placé à côté du fichier de projet d'application.

Pour chaque génération, le fichier d'historique stocke :

- le nom et le chemin d'accès complet du fichier de projet
- le nom et le chemin d'accès complet de la base de données
- la date et l'heure de la génération
- une information indiquant si la génération a réussi (succeed) ou échoué (failed)
- la durée de chaque génération (par exemple application monoposte) et la durée totale de l'opération.
- le nombre de générations ayant réussi et échoué.

## Construction d'une application client/serveur

4<sup>e</sup> Dimension 2004 vous permet de générer des applications client/serveur personnalisées, homogènes, multi-plate-formes et avec option de mise à jour automatique.

### Qu'est-ce qu'une application client/serveur ?

Une application client/serveur est issue de la combinaison de trois éléments :

- une base de données 4D compilée,
- l'application 4D Server,
- l'application 4D Client (MacOS et/ou Windows).

Une fois générée, une application client/serveur se compose de deux parties homogènes, la partie Serveur (unique), et la partie Cliente (à installer sur chaque poste client). A titre de comparaison, rappelons qu'un déploiement standard avec 4D Server nécessite l'application 4D Server, le fichier de structure de la base, le fichier de données de la base et l'application 4D Client.

En outre, l'application client/serveur est personnalisée et son maniement est simplifié :

- Pour lancer la partie serveur, l'utilisateur double-clique simplement sur l'application serveur : il n'est pas nécessaire de sélectionner un fichier de structure.
- Pour lancer la partie cliente, l'utilisateur double-clique simplement sur l'application cliente, qui se connecte directement à l'application serveur : il n'est pas nécessaire de choisir une base de données dans une boîte de dialogue de connexion.

Si la connexion échoue, l'utilisateur pourra cependant saisir l'adresse du serveur dans une boîte de dialogue de connexion standard.

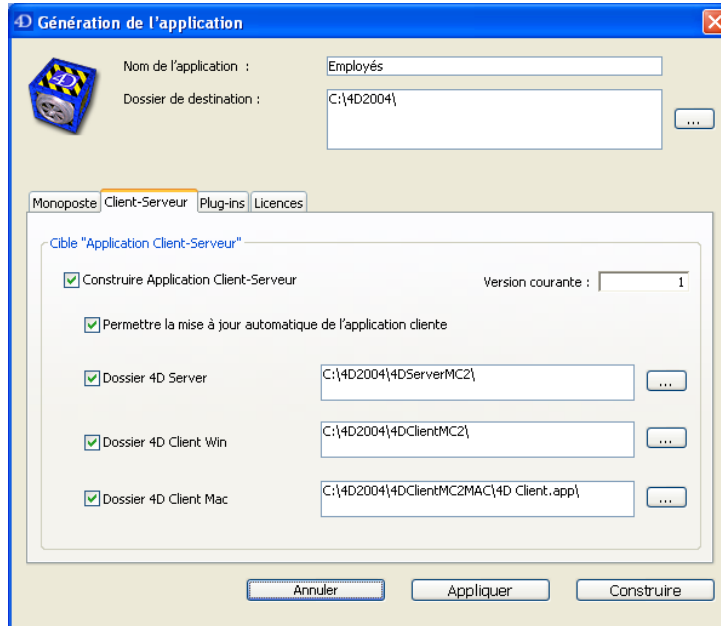
Seule la partie cliente peut se connecter à la partie serveur correspondante. Si un utilisateur tente de se connecter à la partie serveur à l'aide d'une application 4D Client standard, un message d'erreur est retourné et la connexion est impossible.

Enfin, une application client/serveur peut être paramétrée de telle sorte que la partie cliente puisse être mise à jour automatiquement via le réseau. Cette fonction est détaillée dans le [paragraphe "Mise à jour automatique des applications clientes"](#), page 192.



## Paramétrage de l'application

La cible "Application Client-Serveur" est paramétrée dans la page **Client-Serveur** de la boîte de dialogue du générateur d'applications :



- **Construire Application Client-Serveur** : cochez cette option pour indiquer que la génération d'application doit inclure une version client/serveur. Pour que la génération puisse avoir lieu, vous devez ensuite désigner l'emplacement de l'application 4D Server et celui d'au moins une version de l'application 4D Client.
- **Versión courante** : permet d'indiquer le numéro de version courante de l'application générée. Vous pourrez par la suite accepter ou refuser les connexions des applications clientes en fonction de leur numéro de version. L'intervalle de compatibilité des applications clientes et serveur est défini à l'aide de clés XML spécifiques (cf. manuel *4D Clés XML*).
- **Permettre la mise à jour automatique de l'application cliente** : cette option indique que l'application générée prendra en charge le mécanisme de mise à jour des parties clientes via le réseau. Ce mécanisme est détaillé dans le [paragraphe "Mise à jour automatique des applications clientes"](#), page 192.

- **Dossier 4D Server** : cette zone (obligatoire) indique le chemin d'accès du dossier contenant l'application 4D Server que vous souhaitez utiliser pour générer l'application. Cette version doit correspondre à la plate-forme sur laquelle le générateur d'applications est exécuté. Pour sélectionner le dossier de 4D Server, cliquez sur le bouton [...] et utilisez la boîte de dialogue de recherche de dossier pour localiser l'application 4D Server.
- **Dossier 4D Client Win** : cette zone (facultative) indique le chemin d'accès du dossier contenant l'application 4D Client version Windows que vous souhaitez utiliser pour générer l'application. Pour sélectionner le dossier de 4D Client Windows, cliquez sur le bouton [...] et utilisez la boîte de dialogue standard pour localiser l'application. Si vous ne souhaitez pas générer de version Windows pour la partie cliente, ne cochez pas cette option. Vous pouvez utiliser cette option même si vous travaillez sous MacOS. Il n'est pas possible de générer directement la version Windows de l'application cliente à partir de 4<sup>e</sup> Dimension MacOS, toutefois cette option permet de générer l'archive de mise à jour des applications Windows clientes (cf. [paragraphe "Mise à jour automatique des applications clientes"](#), page 192).
- **Dossier 4D Client Mac** : cette zone (facultative) indique le chemin d'accès du dossier contenant l'application 4D Client version MacOS que vous souhaitez utiliser pour générer l'application. Pour sélectionner le dossier de 4D Client MacOS, cliquez sur le bouton [...] et utilisez la boîte de dialogue standard pour localiser l'application. Si vous ne souhaitez pas générer de version MacOS pour la partie cliente, ne cochez pas cette option. Vous pouvez utiliser cette option même si vous travaillez sous Windows. Il n'est pas possible de générer directement la version MacOS de l'application cliente à partir de 4<sup>e</sup> Dimension Windows, toutefois cette option permet de générer l'archive de mise à jour des applications MacOS clientes (cf. [paragraphe "Mise à jour automatique des applications clientes"](#), page 192).

### Personnaliser les dossiers 4D Server et/ou 4D Client

Lors de la construction de l'application client-serveur exécutable, 4<sup>e</sup> Dimension duplique le contenu du dossier 4D Server dans le sous-dossier **Server** du dossier de destination et le contenu du dossier 4D Client dans le sous-dossier **Client** du dossier de destination. Vous pouvez donc parfaitement personnaliser le contenu des dossiers 4D Server et 4D Client d'origine en fonction de vos besoins.

Vous pouvez, par exemple :

- installer une version de 4D Server correspondant à une langue spécifique ;
- ajouter des fichiers ou dossiers dans le dossier **PlugIns** ;
- personnaliser le contenu du dossier 4D Extensions.

---

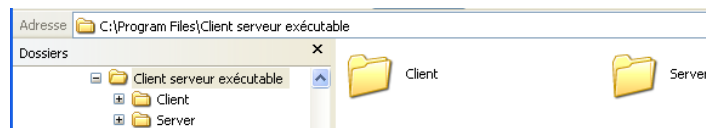
*Note* Sous MacOS, 4D Server est fourni sous forme de progiciel. Vous devrez tout d'abord afficher son contenu (effectuez **Control+clik** sur son icône) afin de pouvoir le modifier.

---

## Fichiers générés

A l'issue du processus de génération d'une application client-serveur, vous devez trouver dans le dossier de destination un nouveau dossier nommé **Client serveur exécutable**.

Ce dossier contient deux sous-dossiers, **Client** et **Server** :



---

*Note* Ces dossiers ne sont pas générés si une erreur est survenue. Dans ce cas, ouvrez le fichier d'historique (cf. [paragraphe "Fichier d'historique", page 183](#)) pour connaître la cause de l'erreur.

---

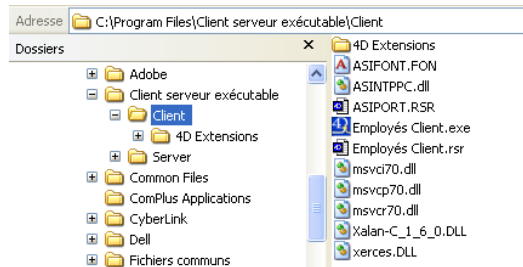
Le dossier **Client** contient la partie cliente de l'application correspondant à la plate-forme d'exécution du générateur d'application. Ce dossier doit être installé sur chaque poste client. Le dossier **Server** contient la partie serveur de l'application.

Le contenu de ces dossiers diffère en fonction de la plate-forme courante :

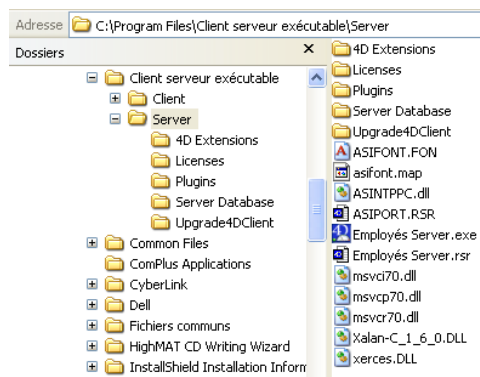
- **Sous Windows**, chaque dossier contient le fichier exécutable de l'application, nommé *NomApplication Client.exe* pour la partie cliente et *NomApplication Server.exe* pour la partie serveur, ainsi que les fichiers *.rsr* correspondants.

Les dossiers contiennent également divers fichiers et dossiers nécessaires au fonctionnement des applications et les éléments personnalisés éventuellement placés dans les dossiers 4D Client et 4D Server d'origine :

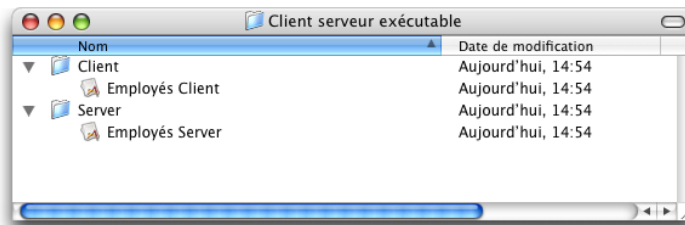
Dossier Client



Dossier Server



- **Sous MacOS**, chaque dossier contient uniquement le progiciel de l'application, nommé *NomApplication Client* pour la partie cliente et *NomApplication Server* pour la partie serveur. Chaque progiciel contient tous les éléments nécessaires à son fonctionnement. Sous MacOS, un progiciel est lancé via un double-clic.

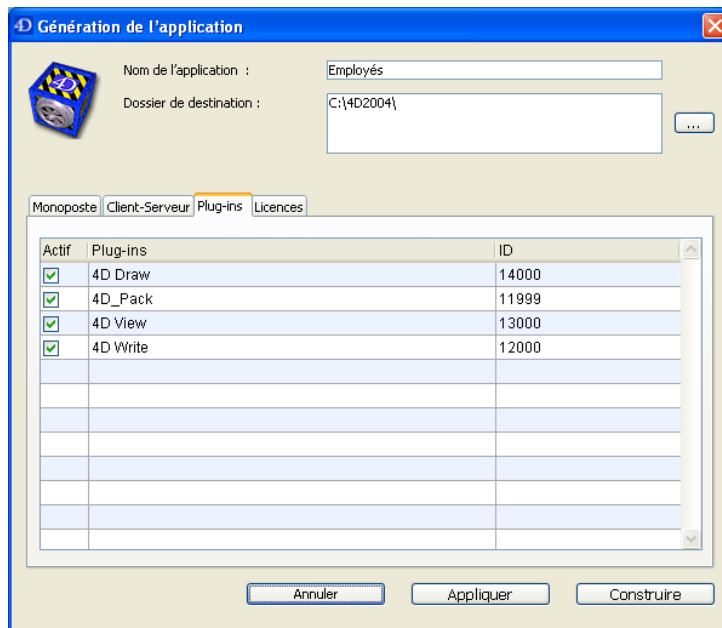


*Note* Les progiciels MacOS générés contiennent les mêmes éléments que les sous-dossiers Windows. Pour les visualiser, vous devrez tout d'abord afficher leur contenu (effectuez **Control+clik** sur son icône) afin de pouvoir le modifier.

Si vous avez coché l'option "Permettre la mise à jour automatique de l'application cliente", un sous-dossier supplémentaire nommé **Upgrade4DClient** est ajouté dans le dossier **Server** (Windows) et le progiciel *NomApplication Server* (MacOS). Ce sous-dossier contient l'application cliente au format MacOS et/ou Windows sous forme de fichier compressé. Ce fichier est utilisé lors de la mise à jour automatique des applications clientes (cf. [paragraphe "Mise à jour automatique des applications clientes"](#), page 192).

*Note* Comme dans les versions précédentes de 4<sup>e</sup> Dimension, le contenu du dossier de destination est supprimé avant toute génération. Ce mécanisme est mis en oeuvre même si vous ne générez qu'une partie de l'application (par exemple, la partie cliente). Par conséquent, veuillez à toujours extraire du dossier de destination les éléments à conserver avant de générer de nouvelles versions.

**Gestion des plug-ins** Le nouveau générateur d'applications vous permet de désigner chaque plug-in que vous souhaitez intégrer dans votre application monoposte ou client-serveur. Pour cela, cliquez sur l'onglet **Plug-ins** afin d'afficher la liste des plug-ins "intégrables" :



Cette liste affiche les plug-ins chargés par l'application 4<sup>e</sup> Dimension courante.

La colonne **Actif** indique les plug-ins qui seront intégrés dans l'application générée. Par défaut, tous les plug-ins sont inclus. Pour désélectionner un plug-in, cochez la case qui lui est associée.

Si vous souhaitez intégrer d'autres plug-ins dans l'application exécutable, il vous suffit de les placer dans un dossier PlugIns ou Win4DX/Mac4DX à côté de l'application 4D Runtime Volume License ou de l'application 4D Server. Le mécanisme de copie du contenu du dossier de l'application source (cf. [paragraphe "Personnaliser les dossiers 4D Server et/ou 4D Client", page 186](#)) permet d'intégrer tout type de fichier à l'application exécutable.

---

*Note* Pour plus d'informations sur l'installation des plug-ins dans 4<sup>e</sup> Dimension 2004, reportez-vous au [paragraphe "Nouvelle architecture des plug-ins", page 18](#).

---

En cas de conflit entre deux versions différentes d'un même plug-in (l'une chargée par 4<sup>e</sup> Dimension et l'autre placée dans le dossier de l'application source), la priorité revient au plug-in installé dans le dossier de 4D Runtime Volume License/4D Server.

---

*Note* L'utilisation de plug-ins dans une version de déploiement nécessite des numéros de licence adéquats.

---

### Gestion des numéros de licence

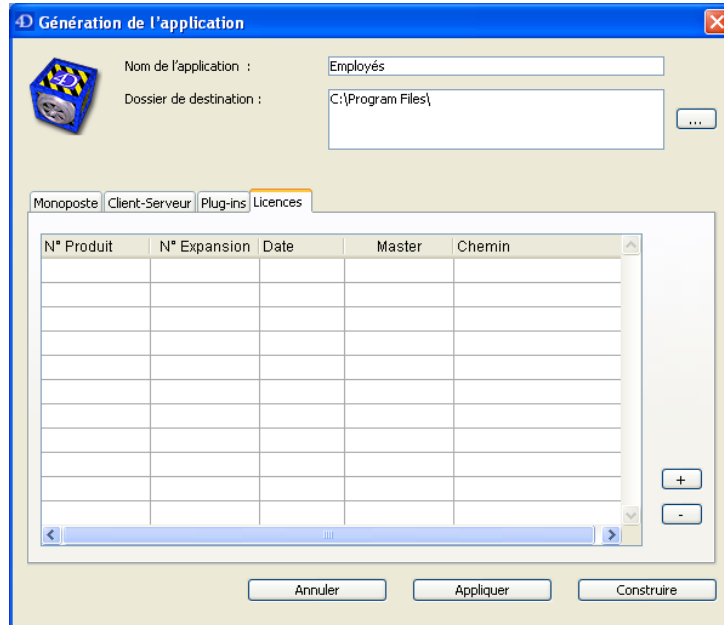
La nouvelle version du générateur d'applications vous permet de spécifier le ou les numéro(s) de licence que vous souhaitez intégrer dans votre application monoposte.

---

*Note* La gestion des licences dans les applications client-serveur exécutables est identique à 4D Server : la saisie des numéros de licence et/ou d'expansion s'effectue dans la boîte de dialogue **Mise à jour des licences** sur le poste serveur.

---

Cliquez sur l'onglet **Licences** afin d'afficher la liste des licences de disponibles :



Par défaut, la liste est vide. Vous devez explicitement ajouter votre licence *Developer Edition* ainsi que chaque licence *4D Runtime Volume License* liée à utiliser dans l'application générée. Vous pouvez ajouter un numéro *Developer Edition* et ses licences associées autres que ceux en cours d'utilisation.

Pour ajouter ou supprimer des licences, utilisez les boutons [+] et [-] situés en bas de la fenêtre.

Lorsque vous cliquez sur le bouton [+], une boîte de dialogue d'ouverture de document apparaît. Vous devez désigner le fichier de sérialisation contenant vos licences de déploiement. Ce fichier, au format HTML, a été généré ou mis à jour au moment de l'acquisition de la licence *Developer Edition* et des licences *4D Runtime Volume License*.

---

*Note* Pour plus d'informations sur l'enregistrement des numéros de licence, reportez-vous au *Guide d'installation* de 4<sup>e</sup> Dimension.

---

Une fois que vous avez sélectionné un fichier, la liste indique les caractéristiques de la licence qu'il contient : numéro de produit ou d'expansion, date de validité, numéro *Developer Edition* lié ("Master") et emplacement. Si la licence est invalide, un message vous le signale.

Vous pouvez désigner autant de fichiers valides que vous voulez. Lors de la génération de l'application exécutable, 4<sup>e</sup> Dimension utilisera les licences les plus appropriées.

À l'issue de la génération, un nouveau fichier de licence de déploiement est automatiquement inclus dans un dossier *Licenses* placé à côté de l'application exécutable (Windows) ou dans le progiciel (MacOS).

### Mise à jour automatique des applications clientes

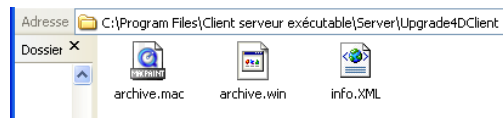
Si vous avez coché l'option **Permettre la mise à jour automatique de l'application cliente** dans la page Client-Serveur du générateur d'applications, les parties clientes de vos applications client/serveur pourront être mises à jour automatiquement via le réseau à chaque nouvelle version de l'application générée.

Ce système évite au développeur de devoir installer "manuellement" les mises à jour sur chaque poste client.

### Comment proposer une mise à jour ?

Dans la pratique, la proposition de mise à jour des applications clientes découle automatiquement de la mise à jour de l'application serveur.

Le principe est le suivant : lors de la génération d'une nouvelle version de l'application client-serveur depuis le générateur d'applications, la nouvelle partie cliente est copiée sous forme compressée dans le sous-dossier **Upgrade4DClient** du dossier **Server** (sous MacOS, ces dossiers sont inclus dans le progiciel serveur). Si vous avez sélectionné les versions Windows et MacOS de 4D Client dans la boîte de dialogue de génération d'application ([paragraphe "Paramétrage de l'application", page 185](#)), un fichier de mise à jour est disponible pour chaque plateforme :



Pour provoquer la mise à jour des applications clientes, il suffit de remplacer l'ancienne version de l'application serveur par la nouvelle puis de l'exécuter. Le reste du processus est automatique.



Côté client, au moment où l'« ancienne » application cliente tente de se connecter à l'application serveur mise à jour, une boîte de dialogue s'affiche sur le poste client, lui indiquant qu'une nouvelle version est disponible. L'utilisateur peut mettre sa version à jour ou annuler la boîte de dialogue.

- Si l'utilisateur clique sur **OK**, la nouvelle version est téléchargée sur le poste client via le réseau. A l'issue du téléchargement, l'ancienne application cliente quitte, la nouvelle est lancée et se connecte au serveur. L'ancienne application est alors placée dans la corbeille de la machine.
- Si l'utilisateur clique sur **Annuler**, la mise à jour est annulée ; si l'ancienne version de l'application cliente n'appartient pas à l'intervalle des numéros de version acceptés par le serveur (cf. paragraphe suivant), l'application quitte et la connexion est impossible. Sinon (par défaut), la connexion est établie.

### Comment forcer la mise à jour ?

Dans certains cas, vous pourrez souhaiter que les applications clientes ne puissent pas annuler le téléchargement des mises à jour. Par exemple, si vous avez utilisé une nouvelle version de l'application source 4D Server/4D Client, il est impératif que la nouvelle version de l'application cliente soit installée sur chaque poste client.

Pour forcer la mise à jour, il vous suffit d'exclure les versions courantes des applications clientes (N-1 et précédentes) de l'intervalle des numéros de version compatibles avec l'application serveur. Dans ce cas, le mécanisme de mise à jour n'autorisera pas la connexion des applications clientes non mises à jour. Par exemple, si la nouvelle version de l'application client-serveur est 6, vous pouvez stipuler que toute application cliente ayant un numéro de version strictement inférieur à 6 ne sera pas autorisé à se connecter.

Le numéro de version courante est défini dans la page **Client-serveur** du générateur d'application (cf. [paragraphe "Paramétrage de l'application"](#), page 185).

Les intervalles de numéros autorisés sont définis dans le projet d'application via des clés XML spécifiques. Pour plus d'informations, reportez-vous au [paragraphe "Boîte de dialogue et projet de génération d'application"](#), page 182 et au manuel *Clés XML de 4D*.

## Web Services

La prise en charge des Web Services a été renforcée dans 4<sup>e</sup> Dimension 2004, du côté serveur et du côté client.

- Côté serveur, 4<sup>e</sup> Dimension publie désormais des méthodes sous forme de Web Services en mode DOC.
- Côté client, 4<sup>e</sup> Dimension 2004 facilite l'utilisation des Web Services publiés en mode DOC, via notamment le traitement des types composés.

### Publication en mode DOC (serveur)

4<sup>e</sup> Dimension 2004 permet de publier des méthodes comme Web Services en mode DOC. Ce mode de publication est requis par certaines applications clientes.

La prise en charge du mode de publication DOC est effectuée par l'application 4<sup>e</sup> Dimension de manière transparente. Les méthodes publiées en tant que Web Services RPC dans la version 2003 de 4<sup>e</sup> Dimension peuvent être publiées telles quelles en mode DOC dans 4<sup>e</sup> Dimension 2004. Aucun développement supplémentaire n'est requis.

Pour publier un Web Service 4<sup>e</sup> Dimension en mode DOC, il suffit d'utiliser l'URL suivant pour le WSDL :

**http://AdresseServeur/4DWSDL/DOC**

Le serveur 4<sup>e</sup> Dimension se charge de traiter les requêtes et les réponses en DOC. Les requêtes DOC sont automatiquement envoyées à l'adresse <http://AdresseServeur/4DSOAP/DOC> (et non plus <http://AdresseServeur/4DSOAP>).

---

*Note* Il est toujours possible d'accéder aux Web Services publiés par 4<sup>e</sup> Dimension en mode RPC via l'URL <http://AdresseServeur/4DWSDL>

---

### Licence SOAP

La publication de Web Services avec 4<sup>e</sup> Dimension 2004 requiert désormais une licence spécifique. Dans les versions précédentes de 4D, la licence Web complète était requise.

Pour permettre de connaître la disponibilité de cette licence à l'aide de la commande Licence disponible, deux constantes ont été ajoutées dans le thème "Licence disponible" :

Constante	Type	Valeur
Licence 4D SOAP	Entier long	808465464
Licence SOAP 4D Client	Entier long	808465465

### Utilisation des Web Services (client)

Deux nouveautés facilitent l'accès aux Web Services dans 4<sup>e</sup> Dimension 2004 (côté client) :

- la prise en charge avancée des Web Services publiés en mode DOC
- la possibilité de définir des options dans les requêtes SOAP adressées aux serveurs.

### Prise en charge avancée du mode DOC

L'utilisation de Web Services publiés en mode DOC est facilitée dans 4<sup>e</sup> Dimension 2004. En particulier, l'Assistant Web Services prend en charge les types composés (ou *complexes*) et leur traduction en code 4D dans les méthodes proxy de manière plus complète.

Désormais, lorsque vous générez via l'Assistant la méthode proxy d'un Web Service publié en mode DOC, un traitement expert est effectué sur les données xml composées de la requête.

A l'aide des nouvelles commandes de gestion des arbres XML, l'Assistant Web Services tentera d'extraire des valeurs exploitables depuis les données xml composées.

Voici un exemple de méthode proxy générée dans 4<sup>e</sup> Dimension 2004 pour un Web Service en mode DOC :

```

` proxy_GetWhois
` http://ws.cdyne.com/whoisquery/whois.asmx?wsdl
` Méthode générée automatiquement par l'assistant Web Services de 4D.
-----
C_TEXTE($1)
C_TEXTE($2)
C_TEXTE($0)
C_ALPHA(16;$!_root)
C_ALPHA(16;$!_subelem)
C_TEXTE($!_namespace)
$!_namespace="http://ws.cdyne.com/"
$!_root=DOM Créer ref XML("GetWhois",$!_namespace)

$!_subelem=DOM Créer élément XML($!_root,"/GetWhois/Query")
DOM ECRIRE VALEUR ELEMENT XML($!_subelem;$1)
$!_subelem=DOM Créer élément XML($!_root,"/GetWhois/LicenseKey")
DOM ECRIRE VALEUR ELEMENT XML($!_subelem;$2)

FIXER PARAMETRE WEB SERVICE("XMLIn";$!_root)
APPELER WEB SERVICE("http://ws.cdyne.com/whoisquery/whois.asmx";"http://ws.cdyne.com/GetWhois";"GetWhois";"http://ws.cdy
DOM FERMER XML($!_root)

Si (OK<=1)
  C_BLOB($!_blob)
  C_ALPHA(16;$!_resroot)
  C_ALPHA(16;$!_ressubelem)
  LIRE RESULTAT WEB SERVICE($!_blob;"XMLOut";*)
  $!_resroot=DOM Analyser variable XML($!_blob)
  $!_ressubelem=DOM Chercher élément XML($!_resroot,"/GetWhoisResponse/GetWhoisResult")
  DOM LIRE VALEUR ELEMENT XML($!_ressubelem;$0)
  DOM FERMER XML($!_resroot)
Fin de si

```

Exemples d'utilisation  
des nouvelles  
commandes XML

L'appel de la méthode proxy au sein de votre base de données est identique à celle de la version précédente de 4<sup>e</sup> Dimension.

**Attention** : seuls les tableaux et les données de type composé sur un niveau (un seul niveau hiérarchique dans la requête SOAP) sont pleinement pris en charge par l'Assistant Web Services. Si des éléments plus complexes sont présents dans la requête, l'Assistant le signalera lors de la découverte du WSDL par un drapeau affiché à côté du nom de la méthode. La prise en charge de ce type de Web Service nécessite généralement des traitements personnalisés de la part du développeur.

**Définition d'options dans les requêtes**

4<sup>e</sup> Dimension 2004 vous permet de définir des options pour les requêtes adressées aux serveurs. Vous pouvez désormais spécifier :

- le header de la requête SOAP,
- le timeout http de la connexion,
- la version de SOAP prise en charge.

Ces options sont prises en charge par la nouvelle commande **FIXER OPTION WEB SERVICE**. Reportez-vous à sa description dans le chapitre *Langage* pour plus d'informations.

# 5

# Module de sauvegarde intégré

## Présentation

4<sup>e</sup> Dimension 2004 inclut un module complet de sauvegarde des bases de données et de récupération en cas d'incident.

Ce module fonctionne avec 4<sup>e</sup> Dimension monoposte et avec 4D Server. Il permet de sauvegarder une base de données en cours d'exploitation, sans qu'il soit nécessaire de quitter l'application. Chaque sauvegarde peut inclure le fichier de structure, le fichier de données (ainsi que ses éventuels segments) et tout fichier ou dossier additionnel. Ces paramètres sont définis au préalable dans les Préférences de l'application.

Les sauvegardes peuvent être déclenchées manuellement ou automatiquement, à intervalles réguliers et sans intervention de l'utilisateur. Des commandes de langage spécifiques ainsi que de nouvelles méthodes base permettent d'intégrer les fonctions de sauvegarde à une interface personnalisée.

La restitution d'une base de données après incident peut s'effectuer automatiquement lors de l'ouverture d'une base endommagée.

En outre, le module de sauvegarde intégré tire parti du fichier d'historique. Ce fichier conserve une trace de chaque opération effectuée sur les données et assure ainsi une sécurité totale entre deux sauvegardes. En cas d'incident sur une base de données en cours d'exploitation, les opérations éventuellement manquantes dans le fichier de données sont automatiquement réintégrées lors de l'ouverture suivante de la base. Une fenêtre spécifique vous permet de visualiser à tout moment le contenu du fichier d'historique.

Le module de sauvegarde intégré vous permet notamment de :

- déclencher à tout moment une sauvegarde complète des fichiers de la base (structure, données, historique, fichiers joints...),
- mettre en place des sauvegardes automatiques à intervalles réguliers — sur la base d'heures, jours, semaines ou mois,
- définir des paramètres avancés pour les sauvegardes (nombre de jeux, compression des fichiers, options de redémarrage après restitution...),
- restituer automatiquement une base de données et ses fichiers joints en cas d'incident,
- intégrer automatiquement dans une base restituée des opérations manquantes mais conservées dans le fichier d'historique,
- revenir en arrière parmi les opérations effectuées sur les données de la base.

## Gestion des sauvegardes

La sauvegarde est la fonction élémentaire du nouveau module de 4<sup>e</sup> Dimension. Sauvegarder la base consiste à générer à un instant donné une copie de la base et de tous les fichiers qui lui sont nécessaires. Cette copie est placée dans un ou plusieurs fichier(s) de sauvegarde.

Tout fichier de sauvegarde pourra être ultérieurement ouvert par 4<sup>e</sup> Dimension en cas d'incident ayant endommagé la base courante ; la base sera alors restituée dans son état antérieur (au moment de la copie).

### Effectuer une sauvegarde

Chaque sauvegarde est effectuée en tenant compte des paramètres définis dans les Préférences de l'application.

Les Préférences permettent de configurer tous les aspects des sauvegardes :

- fichiers à inclure dans la sauvegarde (données / historique, structure, structure utilisateur et fichiers joints),
- emplacement des fichiers de sauvegarde (fichier de sauvegarde principal et fichier de sauvegarde de l'historique),
- gestion du fichier d'historique,

- périodicité des sauvegardes,
- options de sauvegarde : nombre et rotation des jeux de sauvegarde, prise en charge des transactions ou indexations, gestion des échecs, segmentation, compression et intégrité des fichiers de sauvegarde,
- paramétrage des options de restitution automatiques.

Ces paramètres sont fixés à des valeurs par défaut correspondant à une utilisation standard. Leur modification est donc facultative. Pour plus d'informations sur ces paramètres, reportez-vous au [paragraphe "Configuration des fichiers de sauvegarde"](#), page 203 et au [paragraphe "Paramètres de sauvegarde"](#), page 208.

### Déclencher une sauvegarde

Dans 4<sup>e</sup> Dimension 2004, une sauvegarde peut être déclenchée de trois manières :

- manuellement, via la commande **Sauvegarder...** du menu **Fichier** de 4<sup>e</sup> Dimension (mode Utilisation) et 4D Server.
- automatiquement, via le programmeur paramétrable dans les Préférences de l'application,
- par programmation, à l'aide de la commande **SAUVEGARDER**.

Le choix du mode dépend de votre utilisation de la base et de votre stratégie de sauvegarde.

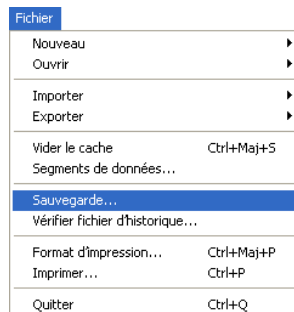
---

*4D Server* Il est possible de déclencher "manuellement" une sauvegarde depuis 4D Client, via une méthode appelant la commande SAUVEGARDER. Dans tous les cas, la commande sera exécutée sur le serveur.

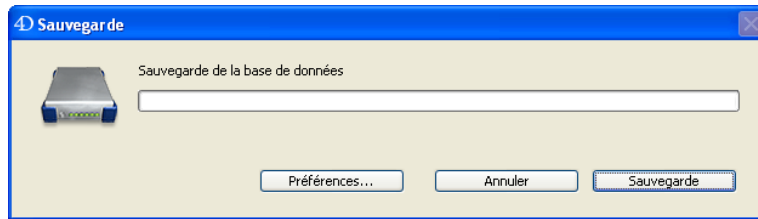
---

► Pour effectuer une sauvegarde manuelle :

- 1 Choisissez la commande **Sauvegarder...** dans le menu **Fichier** de 4<sup>e</sup> Dimension (en mode Utilisation) ou de 4D Server :



La fenêtre de sauvegarde s'affiche :

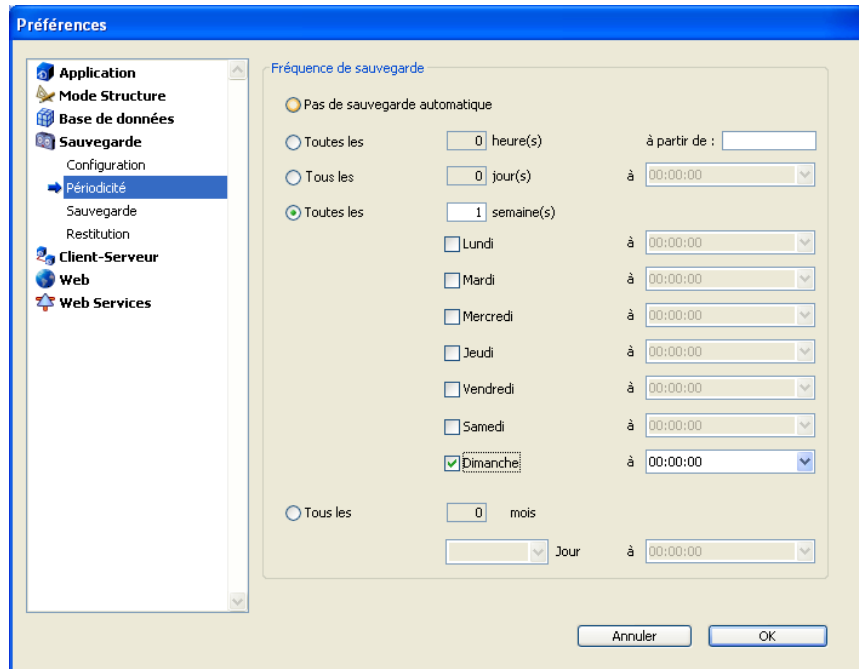


Le bouton **Préférences** provoque l'affichage des préférences générales de sauvegarde (cf. [paragraphe "Configuration des fichiers de sauvegarde"](#), page 203). Le bouton **Annuler** annule la sauvegarde.

2 Cliquez sur le bouton **Sauvegarde** pour déclencher la sauvegarde avec les paramètres courants.

► Pour effectuer une sauvegarde automatique périodique :

1 Dans la page **Périodicité** du thème "Sauvegarde" des Préférences, définissez la fréquence à laquelle vous souhaitez que les sauvegardes soient effectuées :



Les sauvegardes seront effectuées automatiquement au moment défini, sans intervention de l'utilisateur.



*Note* Pour plus d'informations sur le fonctionnement de cette boîte de dialogue, reportez-vous au [paragraphe "Paramétrer des sauvegardes périodiques"](#), page 214.

- Pour effectuer une sauvegarde programmée à l'aide du langage de 4<sup>e</sup> Dimension :

**1 Dans une méthode, exécutez la commande SAUVEGARDER.**

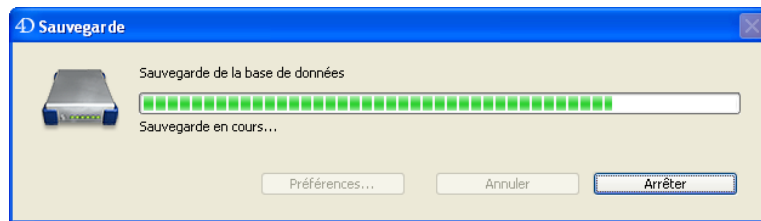
La sauvegarde est alors déclenchée avec les paramètres courants.

Vous pouvez utiliser les nouvelles méthodes bases Sur démarrage sauvegarde et Sur arrêt sauvegarde afin de contrôler le déroulement de la sauvegarde.

Pour plus d'informations sur ce point, reportez-vous au [paragraphe "Sauvegarde"](#), page 370

### Exécution de la sauvegarde

Une fois qu'une sauvegarde a été déclenchée, 4<sup>e</sup> Dimension affiche une boîte de dialogue indiquant la progression de l'opération :



Le bouton **Arrêter** permet à l'utilisateur d'interrompre la sauvegarde à tout instant (cf. [paragraphe "En cas de problème au cours de la sauvegarde"](#), page 202).

Le compte-rendu de la dernière sauvegarde (succès ou échec) est stocké dans la zone "Statut" de la page **Configuration** des Préférences ou dans la fenêtre principale de 4D Server (cf. [paragraphe "Informations sur la dernière sauvegarde"](#), page 207). Il est également enregistré dans Journal des sauvegardes de la base (cf. [paragraphe "Le journal des sauvegardes"](#), page 236).

### Accès à la base durant la sauvegarde

Pendant la sauvegarde, les possibilités d'accès à la base sont les suivantes :

- avec 4<sup>e</sup> Dimension version monoposte, la base est verrouillée en lecture et en écriture, tous les process sont gelés. Toute action est alors impossible.

- avec 4D Server, la base est verrouillée uniquement en écriture, les postes clients peuvent consulter les données. Si un poste client envoie une requête d'ajout, de suppression ou de modification au serveur, il obtient une fenêtre l'invitant à attendre la fin de la sauvegarde. Une fois la base sauvegardée, la fenêtre disparaît d'elle-même et l'action est effectuée. Pour annuler la requête en cours et ne pas avoir à attendre la fin de la sauvegarde, il suffit de cliquer sur le bouton **Annuler l'opération**.

Cependant, si l'action en attente provient d'une méthode lancée avant la sauvegarde, il est déconseillé de l'annuler car seules les opérations restant à effectuer seront annulées. Or, une méthode "à moitié" exécutée peut conduire à des incohérences logiques dans la base.

---

*Note* Lorsque l'action en attente provient d'une méthode et que l'utilisateur clique sur le bouton **Annuler l'opération**, 4D Server renvoie l'erreur -9976 (*Cette commande ne peut être exécutée car la base est en cours de sauvegarde*).

---

### En cas de problème au cours de la sauvegarde

Il est possible que la sauvegarde ne s'effectue pas correctement. Les causes de l'échec peuvent être diverses : interruption par l'utilisateur, fichier joint introuvable, disque de destination défaillant, transaction non terminée, etc. Le traitement de l'incident par 4<sup>e</sup> Dimension diffère en fonction de la cause.

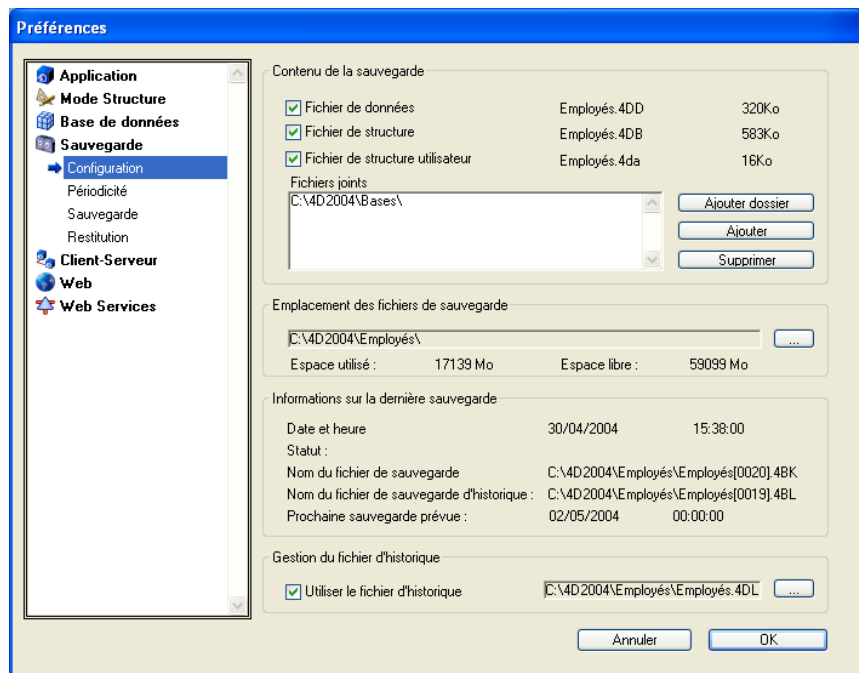
Dans tous les cas, le statut de la dernière sauvegarde (succès ou échec) est affiché dans la page **Configuration** des Préférences de l'application, dans la fenêtre de 4D Server et dans le Journal des sauvegardes (cf. [paragraphe "Informations sur la dernière sauvegarde"](#), page 207).

- **Interruption par l'utilisateur** : le bouton **Arrêter** de la boîte de dialogue de progression de la sauvegarde permet aux utilisateurs d'interrompre la sauvegarde à tout instant. Dans ce cas, la copie des éléments est stoppée et une erreur spécifique est générée. Vous pouvez intercepter cette erreur dans la [Méthode base Sur arrêt sauvegarde](#).
- **Fichier joint introuvable** : lorsqu'un fichier joint est introuvable, 4<sup>e</sup> Dimension effectue une sauvegarde partielle (sauvegarde des fichiers de la base et des fichiers joints accessibles) et retourne une erreur.

- **Sauvegarde impossible** (disque plein ou protégé en écriture, disque manquant, panne du disque, transaction non terminée, base non lancée au moment d'une sauvegarde automatique périodique, etc.) :
  - s'il s'agit du premier échec, 4<sup>e</sup> Dimension effectuera ultérieurement une seconde tentative. Le délai d'attente entre les deux tentatives est défini dans la page **Sauvegarde** des Préférences (cf. [paragraphe "Si la sauvegarde échoue"](#), page 212).
  - si la seconde tentative échoue également, une boîte de dialogue d'alerte système est affichée et une erreur est générée. Vous pouvez intercepter cette erreur dans la **Méthode base Sur arrêt sauvegarde**.
  - les sauvegardes bloquées à cause d'une transaction ou indexation non terminée sont prises en charge via un mécanisme spécifique (cf. [paragraphe "Si transactions actives ou opérations d'index"](#), page 211).

## Configuration des fichiers de sauvegarde

La page **Configuration** des Préférences de l'application vous permet de désigner les fichiers à sauvegarder, l'emplacement des fichiers de sauvegarde et le fichier d'historique. Elle fournit également des informations sur la dernière sauvegarde :



*4D Server* Ces paramètres peuvent être définis depuis le poste 4D Server uniquement.

---

Ces paramètres sont spécifiques à chaque base de données ouverte par l'application 4<sup>e</sup> Dimension.

### Contenu de la sauvegarde

Cette zone permet de définir les fichiers et/ou dossiers à copier lors de la prochaine sauvegarde.

La partie supérieure de la zone liste les fichiers 4<sup>e</sup> Dimension de la base et indique leur taille courante. Vous devez désigner chaque fichier à inclure dans la sauvegarde en cochant l'option correspondante. Une option grisée signale que le fichier correspondant n'est pas disponible dans la base. Vous pouvez sélectionner les fichiers 4<sup>e</sup> Dimension que vous voulez, en fonction de leur fréquence de modification, leur intérêt stratégique, leur taille, etc. Aucun fichier n'est obligatoire. La partie inférieure liste les chemins d'accès des éventuels fichiers joints à la sauvegarde.

■ **Fichier de données** : fichier de données de la base.

Si la base contient plusieurs segments, seuls le nom et la taille du premier segment sont affichés. Bien entendu, tous les segments seront sauvegardés.

Lorsque cette option est cochée, le **fichier d'historique** courant de la base, s'il existe, est sauvegardé en même temps que les données. La sauvegarde provoque la fermeture et la sauvegarde du fichier d'historique courant puis la création d'un nouveau fichier d'historique vierge. Ce fonctionnement évite que la taille du fichier d'historique n'augmente de manière excessive.

Pour plus d'informations sur le fichier d'historique, reportez-vous au [paragraphe "Gestion du fichier d'historique"](#), page 217.

■ **Fichier de structure** : fichier de structure de la base.

Dans le cas de bases compilées et fusionnées avec 4D Runtime Volume License (cf. [Nouveaux noms pour les applications de déploiement](#)), cette option permet de sauvegarder le fichier .exe sous Windows et le progiciel sous MacOS.

■ **Fichier de structure utilisateur** : fichier de structure utilisateur de la base, contenant les formulaires utilisateurs personnalisés. Pour plus d'informations sur ce fichier, reportez-vous au [paragraphe "Formulaires utilisateurs"](#), page 254.

- **Fichiers joints** : cette zone permet de désigner un ensemble de fichiers et/ou de dossiers à sauvegarder en même temps que la base. Ces fichiers peuvent être de tout type (documents ou modèles de plug-ins, étiquettes, états, images, etc.).  
Vous pouvez désigner soit des fichiers individuels, soit des dossiers dont le contenu sera intégralement sauvegardé. Chaque élément joint est listé avec son chemin d'accès complet dans la zone "Fichiers joints".
- **Ajouter dossier** : lorsque vous cliquez sur ce bouton, 4<sup>e</sup> Dimension affiche une boîte de dialogue permettant de sélectionner un dossier à joindre à la sauvegarde. En cas de restitution, le dossier sera récupéré avec sa structure interne. Vous pouvez désigner tout dossier ou volume connecté au poste, à l'exception du dossier contenant les fichiers de la base.
- **Ajouter** : lorsque vous cliquez sur ce bouton, 4<sup>e</sup> Dimension affiche une boîte de dialogue permettant de sélectionner un fichier à joindre à la sauvegarde.  
Il n'est pas possible de désigner un fichier de la base comme fichier joint.
- **Supprimer** : permet de retirer de la liste des fichiers joints l'élément sélectionné.

---

*Note* Il est possible qu'un ou plusieurs fichiers joints ne soient pas accessibles au moment de la sauvegarde (nom ou chemin d'accès modifié, volume éteint, etc.). Dans ce cas, la sauvegarde sera effectuée sans le(s) fichier(s) manquant(s) et une erreur sera générée. Vous pouvez intercepter cette erreur dans la [Méthode base Sur arrêt sauvegarde](#). L'erreur est également indiquée dans le Journal des sauvegardes.

---

### Emplacement des fichiers de sauvegarde

Cette zone permet de définir l'emplacement auquel seront stockés les fichiers de sauvegarde ainsi que les fichiers de sauvegarde du fichier d'historique (le cas échéant).

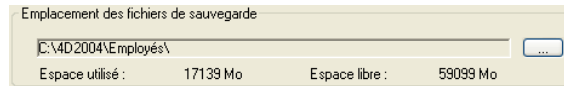
4<sup>e</sup> Dimension génère deux types de fichiers de sauvegarde : les fichiers de sauvegarde et les fichiers de sauvegarde de l'historique. Les fichiers de sauvegarde sont particuliers, et ce à double titre :

- un fichier de sauvegarde peut contenir plusieurs fichiers (structure, données, fichiers joints, etc.),

- ils sont sécurisés par des mécanismes de vérification internes (ces mécanismes peuvent être configurés via les paramètres avancés de la sauvegarde).

Par défaut, 4<sup>e</sup> Dimension sauvegarde ces fichiers à côté du fichier de données de la base. Il est cependant fortement conseillé de désigner un emplacement situé sur un autre disque afin de réduire le risque de perte de données en cas de panne du disque dur contenant la base et les sauvegardes.

Pour modifier l'emplacement auquel ces fichiers devront être enregistrés, cliquez sur le bouton [...] :



Une boîte de dialogue de sélection de dossier apparaît, vous permettant de désigner un dossier ou un volume devant accueillir les sauvegardes. Les zones "Espace utilisé" et "Espace libre" sont automatiquement mises à jour et indiquent l'espace disque disponible sur le volume du dossier sélectionné.

Vous devez veiller à ce que l'espace libre soit suffisant pour accueillir l'ensemble des sauvegardes. Dans le cas où une sauvegarde échoue par manque de place sur le volume de stockage, une erreur est générée. Vous pouvez intercepter cette erreur dans la [Méthode base Sur arrêt sauvegarde](#). L'erreur est également indiquée dans la zone d'information et dans le Journal des sauvegardes.

### Noms des fichiers de sauvegarde

4<sup>e</sup> Dimension nomme les fichiers de sauvegarde suivant une nomenclature spécifique sur laquelle s'appuient les fonctions automatiques de restitution. Il n'est pas possible de modifier cette nomenclature.

- Les sauvegardes standard sont nommées **Nombase[xxxx].4BK**, où *nombase* est le nom du fichier de données de la base et *xxxx* représente le numéro de la sauvegarde. Par exemple, la 26<sup>e</sup> sauvegarde de la base Factures sera nommée *Factures[0026].4BK*.  
Si la sauvegarde est segmentée, 4<sup>e</sup> Dimension ajoute le numéro de segment sous la forme *-xxxx*. Par exemple, le 3<sup>e</sup> segment de la 26<sup>e</sup> sauvegarde de la base Factures sera nommé *Factures[0026-0003].4BK*.  
Pour plus d'informations sur les segments, reportez-vous au [paragraphe "Archive"](#), page 213.

- Les sauvegardes des fichiers d'historique sont nommées **Nomhistorique[xxxx].4BL**, où *nomhistorique* est le nom du fichier d'historique de la base et *xxxx* représente le numéro de la sauvegarde (à partir de 0). Par exemple, la 13<sup>e</sup> sauvegarde du fichier d'historique Histo sera nommée *Histo[0012].4BL*.

Si la sauvegarde de l'historique est segmentée, 4<sup>e</sup> Dimension ajoute le numéro de segment sous la forme -*xxxx*. Par exemple, le 2<sup>e</sup> segment de 13<sup>e</sup> sauvegarde du fichier d'historique Histo sera nommé *Histo[0012-0002].4BL*. Pour plus d'informations sur les segments, reportez-vous au [paragraphe "Archive", page 213](#).

A noter que les sauvegardes du fichier d'historique débutent à 0 alors que les sauvegardes des fichiers de la base débutent à 1. En effet, à la première sauvegarde des fichiers de la base (*backup[0001].4BK* par exemple), la sauvegarde de l'historique sera nommée *histo[0000]* : elle représente les modifications apportées au fichier de données à partir de son état "vide" et ne peut être intégrée que dans un fichier de données vide. Par conséquent, une sauvegarde d'historique nommée par exemple *histo[0025].4BL* doit être interprétée comme "26<sup>e</sup> sauvegarde de l'historique, correspondant aux opérations effectuées entre la 25<sup>e</sup> et la 26<sup>e</sup> sauvegarde de la base". *histo[0025].4BL* correspond donc à la sauvegarde *backup[0025].4BK*.

- Note* Les intervalles de numérotation des sauvegardes sont les suivants :
- sauvegarde : de 1 à 9999
  - sauvegarde de l'historique : de 0 à 9998
  - segment : de 1 à 9999

### Informations sur la dernière sauvegarde

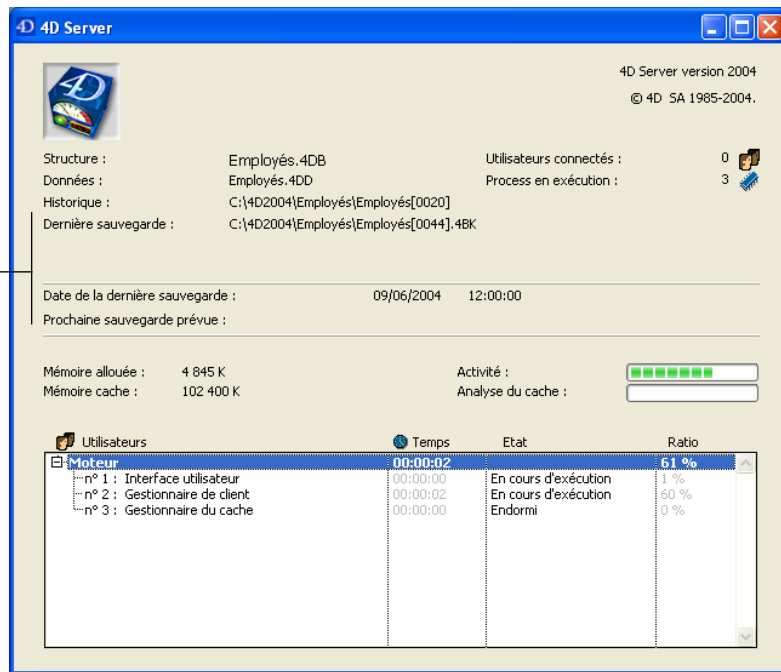
La zone "Informations sur la dernière sauvegarde" permet de connaître le déroulement de la dernière sauvegarde de la base. La zone est renseignée si au moins une sauvegarde a eu lieu.

- **Date et heure** : date et heure de la dernière sauvegarde.
- **Statut** : cette zone affiche le code d'erreur de la dernière sauvegarde ainsi qu'une description de ce code. Si la sauvegarde s'est déroulée correctement, le code 0 (*sauvegarde réussie*) est affiché. Dans le cadre d'un système de sauvegardes périodiques, vous pouvez utiliser cette zone pour contrôler que la dernière sauvegarde a bien eu lieu.

- **Nom du fichier de sauvegarde** : chemin d'accès et nom du fichier de la dernière sauvegarde principale.  
Si la sauvegarde est segmentée, le nom du premier segment est affiché.
- **Nom du fichier de sauvegarde d'historique** : chemin d'accès et nom du fichier de la dernière sauvegarde de l'historique (le cas échéant).
- **Prochaine sauvegarde prévue** : date et heure de la prochaine sauvegarde ; cette zone est renseignée si une stratégie de sauvegardes périodiques a été mise en place.

Avec 4D Server, ces informations sont également affichées dans la fenêtre principale du serveur :

Informations sur la dernière sauvegarde



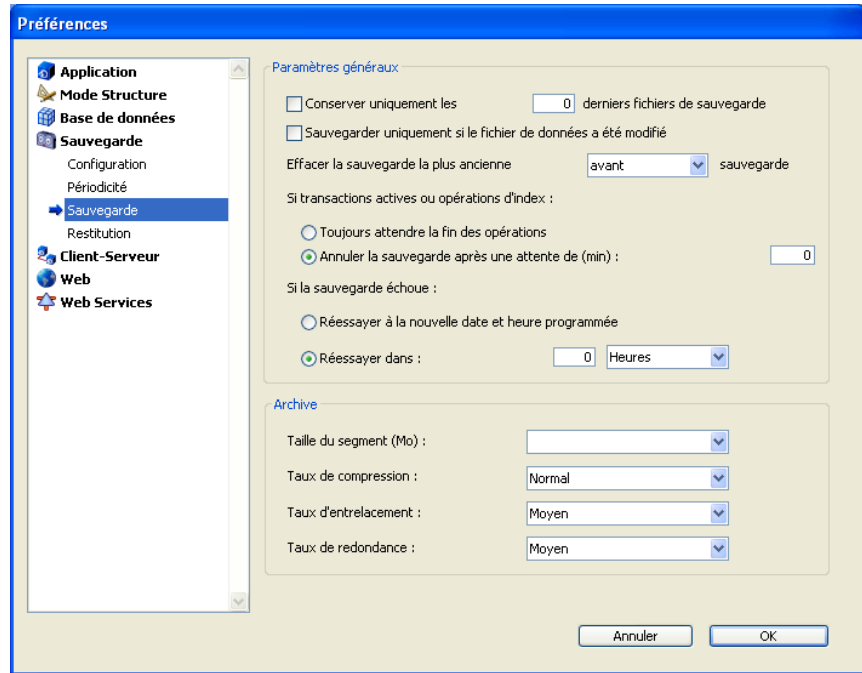
*Note* Les paramètres liés à la gestion du fichier d'historique sont traités dans le [paragraphe "Gestion du fichier d'historique", page 217](#).

## Paramètres de sauvegarde

Comme les paramètres de configuration, les paramètres de sauvegarde sont utilisés à chaque sauvegarde. Toutefois, la modification de ces paramètres est facultative. Leurs valeurs par défaut correspondent à une utilisation standard de la fonction de sauvegarde.



Les paramètres de sauvegarde sont définis dans la page **Sauvegarde** des Préférences de l'application :



*4D Server* Les paramètres de sauvegarde peuvent être définis depuis le poste 4D Server uniquement.

Ces paramètres sont spécifiques à chaque base de données ouverte par l'application 4<sup>e</sup> Dimension.

### Conserver uniquement N derniers fichiers de sauvegarde

Ce paramètre permet d'activer et de configurer le mécanisme de suppression des fichiers de sauvegarde les plus anciens. Ce mécanisme permet de ne conserver que les N derniers fichiers d'archives sur le disque de sauvegarde — le plus ancien fichier étant effacé à chaque nouvelle sauvegarde— et ainsi éviter tout risque de saturation du volume.

Le principe de fonctionnement est le suivant : après avoir terminé la sauvegarde courante, 4<sup>e</sup> Dimension efface l'archive la plus ancienne si celle-ci est localisée au même endroit que l'archive à sauvegarder et porte le même nom (vous pouvez, pour des raisons d'économie de place, demander que l'archive la plus ancienne soit effacée avant la sauvegarde).

Si, par exemple, le nombre de jeux est fixé à 3, les trois premières sauvegardes créent successivement les archives MaBase-0001, MaBase-0002 et MaBase-0003. Lors de la quatrième sauvegarde, l'archive MaBase-0004 est créée alors que l'archive MaBase-0001 est supprimée.

A partir de la place sur disque que vous décidez d'accorder à vos sauvegardes, vous pouvez déterminer le nombre maximum possible de jeux de sauvegarde par la formule suivante :

$$\text{Nombre de jeux maximum} = \frac{\text{Place disponible pour les sauvegardes}}{\text{Taille maximale d'une sauvegarde}} - 1 \text{ jeu}$$

Il faut retrancher 1 au nombre de jeux maximum car par défaut 4<sup>e</sup> Dimension effectue d'abord la sauvegarde courante, puis efface du disque l'archive la plus ancienne. Ce fonctionnement peut cependant être modifié (cf. ci-dessous le paragraphe "Effacer la sauvegarde la plus ancienne avant de sauvegarder").

Par défaut, le mécanisme de suppression des jeux est activé et 4<sup>e</sup> Dimension conserve 3 jeux de sauvegarde.

Pour ne pas activer le mécanisme, désélectionnez l'option.

---

*Note* Ce paramètre concerne à la fois les jeux de sauvegarde de la base et les jeux de sauvegarde de l'historique.

---

**Sauvegarder  
uniquement si le fichier  
de données a été  
modifié**

Lorsque cette option est cochée, 4<sup>e</sup> Dimension déclenche les sauvegardes périodiques uniquement si des données ont été ajoutées, modifiées ou supprimées dans la base depuis la dernière sauvegarde. Dans le cas contraire, la sauvegarde prévue est annulée et reportée à l'échéance suivante. Aucune erreur n'est générée ; le report est toutefois indiqué dans le Journal des sauvegardes.

Cette option permet notamment d'économiser du temps machine sur la sauvegarde de bases principalement utilisées en consultation. A noter que l'activation de cette option ne prend pas en compte les éventuelles modifications apportées au fichier de structure ou aux fichiers joints.

---

*Notes*

- Pour plus d'informations sur les sauvegardes périodiques, reportez-vous au [paragraphe "Paramétrer des sauvegardes périodiques"](#), page 214.
- Lorsque cette option est cochée et que la sauvegarde est déclenchée manuellement, 4<sup>e</sup> Dimension affiche une boîte de dialogue indiquant que le fichier de données n'a pas été modifié et permettant à

l'utilisateur de confirmer ou d'annuler l'opération.

- Ce paramètre concerne à la fois les sauvegardes de la base et les sauvegardes de l'historique.

### Effacer la sauvegarde la plus ancienne avant/après la sauvegarde

Cette option n'est utilisée que si l'option "Garder les N fichiers de sauvegarde" est cochée. Elle vous permet de spécifier si 4<sup>e</sup> Dimension doit commencer par effacer l'archive la plus ancienne avant d'effectuer une sauvegarde (option **avant**) ou si l'effacement doit être effectué après la sauvegarde (option **après**). Pour que ce mécanisme fonctionne, l'archive la plus ancienne ne doit pas avoir été renommée ou déplacée.

L'option **avant** permet une économie de place. Il n'est alors pas nécessaire de retrancher 1 au nombre de sauvegardes possibles, ce qui permet, par exemple, de stocker deux archives de 2 Go sur un disque de 5 Go. A noter qu'en cas d'interruption en cours de sauvegarde, vous ne pourrez accéder ni à l'ancienne sauvegarde, puisqu'elle a été préalablement détruite, ni à la sauvegarde courante qui n'aura pas été terminée. Cependant, la base de données ne pouvant pas être modifiée en cours de sauvegarde, vous avez l'assurance que votre fichier de données est intact, même si vous ne disposez plus d'une sauvegarde valide. Vous vous trouvez en quelque sorte dans la situation de l'automobiliste dont la roue de secours crève, alors que les quatre pneus sont intacts. Une fois le système redémarré et l'archive incomplète détruite, procédez alors immédiatement à une nouvelle sauvegarde.

L'option **après** constitue une sécurité supplémentaire mais nécessite davantage de place sur le volume de sauvegarde.

### Si transactions actives ou opérations d'index

Les transactions et les indexations sont des opérations critiques empêchant la sauvegarde de la base tant qu'elles ne sont pas terminées. Avant de débiter une sauvegarde (quel que soit le mode de déclenchement), 4<sup>e</sup> Dimension ou 4D Server examine la base de données ; si le programme détecte une opération critique en cours, il suspend la procédure de sauvegarde.

Ce paramètre vous permet de définir la durée d'attente dans ce cas :

- **Toujours attendre la fin des opérations** : la sauvegarde est suspendue tant que l'opération n'est pas terminée ; elle est ensuite immédiatement déclenchée. Ce principe garantit que la sauvegarde est bien effectuée mais ne permet pas de contrôler précisément le moment de la sauvegarde.

- **Annuler la sauvegarde après une attente de (min)** : la sauvegarde est suspendue pendant la durée définie afin de laisser le temps à la transaction ou à l'indexation de se terminer. Si à l'issue de ce délai, l'opération critique est toujours en cours, la sauvegarde est considérée comme ayant échoué, une erreur est inscrite dans le Journal des sauvegardes. Le mécanisme défini pour la gestion des échecs prend alors le relais (cf. paragraphe ci-dessous).

---

*Note* Evitez absolument d'ouvrir une transaction au démarrage d'une base de données, transaction qui ne serait validée ou annulée qu'en sortant de la base. Evitez également de présenter au cours d'une transaction des boîtes de dialogue de saisie ou de validation à l'utilisateur. Si ce dernier s'absente au lieu de valider le dialogue, et que la sauvegarde démarre à ce moment, la base ne pourra plus exécuter de nouvelles transactions jusqu'à ce que la boîte de dialogue soit validée.

---

**Si la sauvegarde échoue** Cette option permet de définir le mécanisme de prise en charge des échecs des sauvegardes (sauvegarde impossible).

---

- Notes*
- Tous les incidents ne mènent pas à l'échec de la sauvegarde (cf. [paragraphe "En cas de problème au cours de la sauvegarde", page 202](#)).
  - 4<sup>e</sup> Dimension considère qu'une sauvegarde a échoué si la base n'était pas lancée au moment théorique de la sauvegarde automatique périodique (cf. [paragraphe "Paramétrer des sauvegardes périodiques", page 214](#)).
- 

Lorsqu'une sauvegarde est impossible, 4<sup>e</sup> Dimension permet d'effectuer une nouvelle tentative. Deux options sont possibles :

- **Réessayer à la nouvelle date et heure programmée** : cette option n'a de sens que dans le cadre de sauvegardes automatiques périodiques (cf. [paragraphe "Paramétrer des sauvegardes périodiques", page 214](#)). Elle revient à annuler la sauvegarde ayant échoué. Une erreur est générée.
- **Réessayer dans N heure(s) ou minute(s)** : lorsque cette option est cochée, une nouvelle tentative de sauvegarde est effectuée à l'issue du délai défini. Ce mécanisme permet d'anticiper certaines circonstances bloquant la sauvegarde. Vous pouvez fixer un délai en heures ou en minutes à l'aide du menu correspondant. Si la nouvelle tentative échoue également, une erreur est générée et l'échec est inscrit dans les zones de statut de la dernière sauvegarde et dans le Journal des sauvegardes.

## Archive

Cette zone permet de définir les options de génération des archives. Ces options s'appliquent aux fichiers de sauvegarde principaux et aux fichiers de sauvegarde de l'historique.

### ■ Taille du segment (Mo)

4<sup>e</sup> Dimension vous permet de segmenter les archives, c'est-à-dire de les découper en morceaux de taille fixe. Ce fonctionnement permet par exemple de stocker une sauvegarde sur plusieurs volumes (CDs, ZIPs, etc.). Au moment de la restitution, 4<sup>e</sup> Dimension fusionnera automatiquement les segments. Chaque segment est appelé *MaBase[xxxx-yyy].4BK*, où *xxxx* représente le numéro de la sauvegarde et *yyy* celui du segment. Par exemple, les trois segments de la sixième sauvegarde de la base MaBase seront appelés *MaBase[0006-0001].4BK*, *MaBase[0006-0002].4BK* et *MaBase[0006-0003].4BK*.

Le menu **Taille du segment** est une combo box permettant de définir la taille en Mo de chaque segment de sauvegarde. Vous pouvez choisir une des tailles prédéfinies ou saisir une taille spécifique entre 0 et 2048. Si vous passez 0, aucune segmentation n'est effectuée (équivalent à la valeur **Aucune**).

### ■ Taux de compression

Par défaut, les sauvegardes sont compressées par 4<sup>e</sup> Dimension. Ce principe permet d'économiser de la place sur le disque. En contrepartie, la phase de compression des fichiers peut ralentir sensiblement les sauvegardes dans le cas de la manipulation de gros volumes de données.

L'option **Taux de compression** vous permet d'ajuster la compression :

- **Aucun** : aucune compression n'est appliquée. La sauvegarde peut être sensiblement plus rapide mais les fichiers d'archives sont plus volumineux sur le disque.
- **Normal** (par défaut) : cette option constitue un compromis vitesse de sauvegarde/taille des archives.
- **Elevé** : le taux de compression maximal est appliqué aux archives. Les fichiers d'archives prennent le moins de place possible sur le disque mais la sauvegarde peut être sensiblement ralentie.

### ■ Taux d'entrelacement et Taux de redondance

4<sup>e</sup> Dimension génère des archives à l'aide d'algorithmes spécifiques, basés sur des mécanismes d'optimisation (entrelacement) et de sécurisation (redondance). Vous pouvez paramétrer ces mécanismes en fonction de vos besoins. Pour ces deux options, 4<sup>e</sup> Dimension propose par défaut un taux **Moyen**. Les menus associés à ces options vous permettent de choisir un taux **Faible**, **Elevé** ou **Aucun** taux.

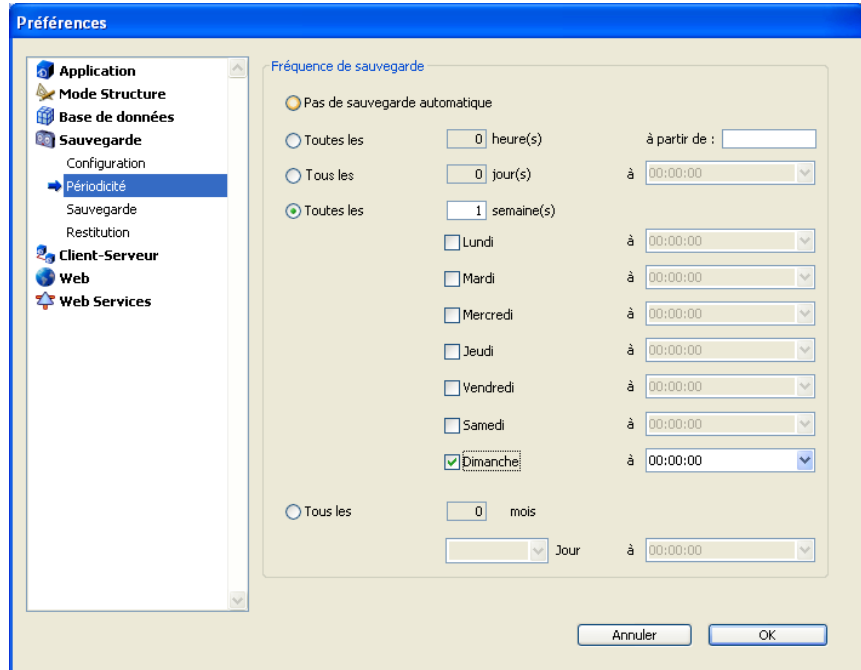
- **Taux d'entrelacement** : l'entrelacement consiste à stocker les données dans des secteurs non contigus, afin d'augmenter la sécurité en cas d'endommagement des secteurs. En contrepartie, plus ce taux est élevé, plus le traitement des données consommera de la mémoire.
- **Taux de redondance** : la redondance permet de sécuriser les données présentes dans un fichier en répétant plusieurs fois les mêmes informations. Plus le taux est élevé, plus le fichier est sécurisé, mais plus le stockage est lent et la taille du fichier importante.

### Paramétrer des sauvegardes périodiques

Vous pouvez automatiser les sauvegardes de vos bases ouvertes avec 4<sup>e</sup> Dimension ou 4D Server (même lorsqu'aucun poste client n'est connecté). Le principe consiste à définir une fréquence de sauvegarde (en heures, jours, semaines ou mois) ; à chaque échéance, 4<sup>e</sup> Dimension déclenche automatiquement une sauvegarde en tenant compte des paramètres de sauvegarde définis dans les Préférences.

Si l'application n'était pas lancée au moment théorique de la sauvegarde, 4<sup>e</sup> Dimension considère au lancement suivant que la sauvegarde a échoué et applique les paramètres adéquats, définis dans les Préférences de la base (cf. [paragraphe "Si la sauvegarde échoue"](#), page 212).

Les paramètres des sauvegardes périodiques sont définis dans la page **Périodicité** des Préférences de l'application :



Vous devez dans un premier temps choisir une échelle de fréquence (jours, semaines...) en cliquant sur le bouton radio correspondant. Vous devez ensuite éventuellement préciser le moment de la sauvegarde.

- **Pas de sauvegarde automatique** : la fonction de sauvegarde périodique est inactivée.
- **Toutes les N heure(s)** : cette option permet de programmer des sauvegardes sur une base horaire. Vous pouvez saisir une valeur comprise entre 1 et 24.
  - **à partir de** : permet de définir l'heure à laquelle débutera la première sauvegarde horaire.
- **Tous les N jour(s) à N** : cette option permet de programmer des sauvegardes sur une base journalière. Saisissez 1 si vous souhaitez une sauvegarde quotidienne. Lorsque vous cochez cette option, vous devez indiquer l'heure à laquelle la sauvegarde doit être déclenchée.

- **Toutes les  $N$  semaine(s), jour à  $N$**  : cette option permet de programmer des sauvegardes sur une base hebdomadaire. Saisissez 1 si vous souhaitez une sauvegarde hebdomadaire. Lorsque vous cochez cette option, vous devez indiquer le ou les jour(s) de la semaine et l'heure à laquelle chaque sauvegarde doit être déclenchée. Vous pouvez cocher un ou plusieurs jour(s) de la semaine. Par exemple, vous pouvez utiliser cette option pour définir deux sauvegardes hebdomadaires : une le mercredi et une le vendredi.
- **Tous les  $N$  mois,  $N^{\text{e}}$  jour à  $N$**  : cette option permet de programmer des sauvegardes sur une base mensuelle. Saisissez 1 si vous souhaitez une sauvegarde mensuelle. Lorsque vous cochez cette option, vous devez indiquer le jour de chaque mois auquel la sauvegarde doit être déclenchée, ainsi que l'heure de déclenchement.

## Sauvegarde automatique

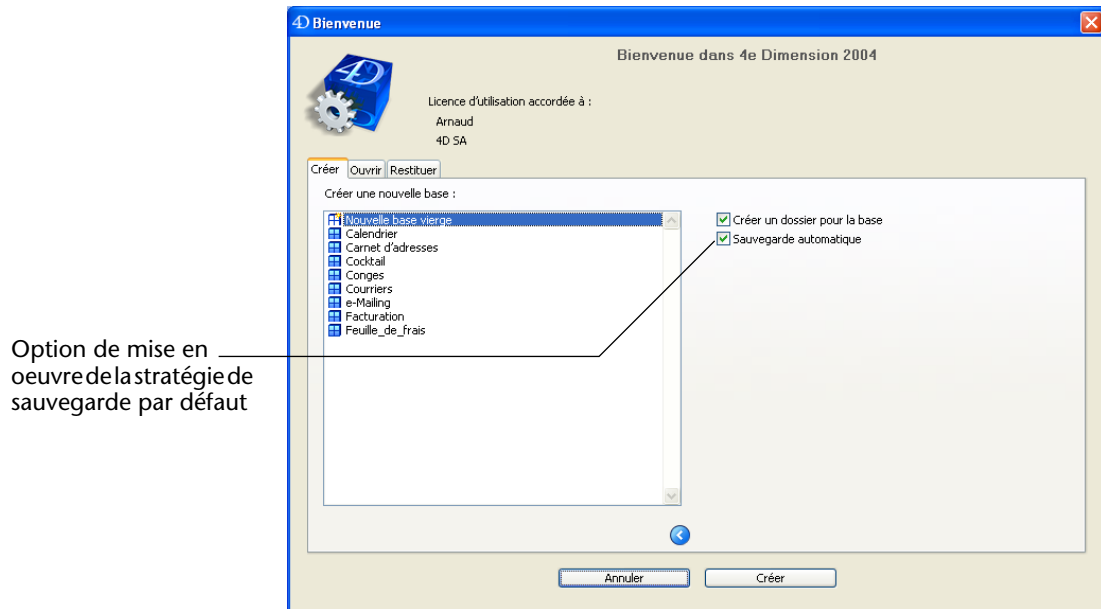
4<sup>e</sup> Dimension propose une stratégie de sauvegarde par défaut au moment de la création d'une base de données. Cette stratégie met en oeuvre un ensemble de paramètres de sauvegarde assurant une sécurité minimale de la base.

Il est bien entendu possible de modifier ces paramétrages après la création de la base.

La stratégie de sauvegarde est activée par défaut à la création d'une nouvelle base. Pour ne pas l'utiliser, il suffit de désélectionner l'option



**Sauvegarde automatique** dans la page d'options de la boîte de dialogue de création de base :



Option de mise en  
oeuvre de la stratégie de  
sauvegarde par défaut

La stratégie de sauvegarde par défaut inclut les paramètres suivants :

- Création et utilisation d'un fichier d'historique (nommé *NomFichier-Données.4dl*).
- Sauvegarde de tous les fichiers de la base (données, historique, structure et structure utilisateur) à côté du fichier de structure de la base.
- Sauvegarde automatique chaque dimanche à 00:00:00 heures.
- Paramètres de sauvegarde par défaut (conservation de 3 jeux de sauvegarde, taux moyens, etc.).
- Toutes les options de restitution automatique.

## Gestion du fichier d'historique

Une base exploitée de manière continue enregistre en permanence des modifications, des ajouts ou des suppressions d'enregistrements. Réaliser des sauvegardes régulières des données est important mais ne permet pas, en cas d'incident, de récupérer les données saisies depuis la dernière sauvegarde. Pour répondre à ce besoin, 4<sup>e</sup> Dimension dispose

d'un outil particulier : le fichier d'historique. Ce fichier permet d'assurer la sécurité permanente des données de la base.

En outre, 4<sup>e</sup> Dimension travaille en permanence avec un *cache* de données situé en mémoire. Toute modification effectuée sur les données de la base est stockée provisoirement dans le cache avant d'être écrite sur le disque dur. Ce principe permet d'accélérer le fonctionnement des applications ; en effet, les accès mémoire sont bien plus rapides que les accès disque. Si un incident survient sur la base avant que les données stockées dans le cache aient pu être écrites sur le disque, vous devrez intégrer le fichier d'historique courant afin de récupérer entièrement la base.

### Présentation

L'historique généré par 4D se présente sous la forme d'un fichier dans lequel toutes les opérations effectuées sur les données d'une base viennent s'inscrire séquentiellement. Ainsi, chaque opération effectuée par un utilisateur provoque deux actions simultanées : une première sur les données de la base (l'instruction est exécutée normalement) et une seconde dans le fichier d'historique (la description de l'opération y est enregistrée). Le fichier d'historique se construit de manière indépendante, sans gêner ni ralentir le travail de l'utilisateur. Une base ne peut travailler qu'avec un seul fichier d'historique à la fois.

Le fichier d'historique enregistre quatre types d'opérations :

- les ajouts d'enregistrements,
- les modifications d'enregistrements,
- les suppressions d'enregistrements,
- les transactions.

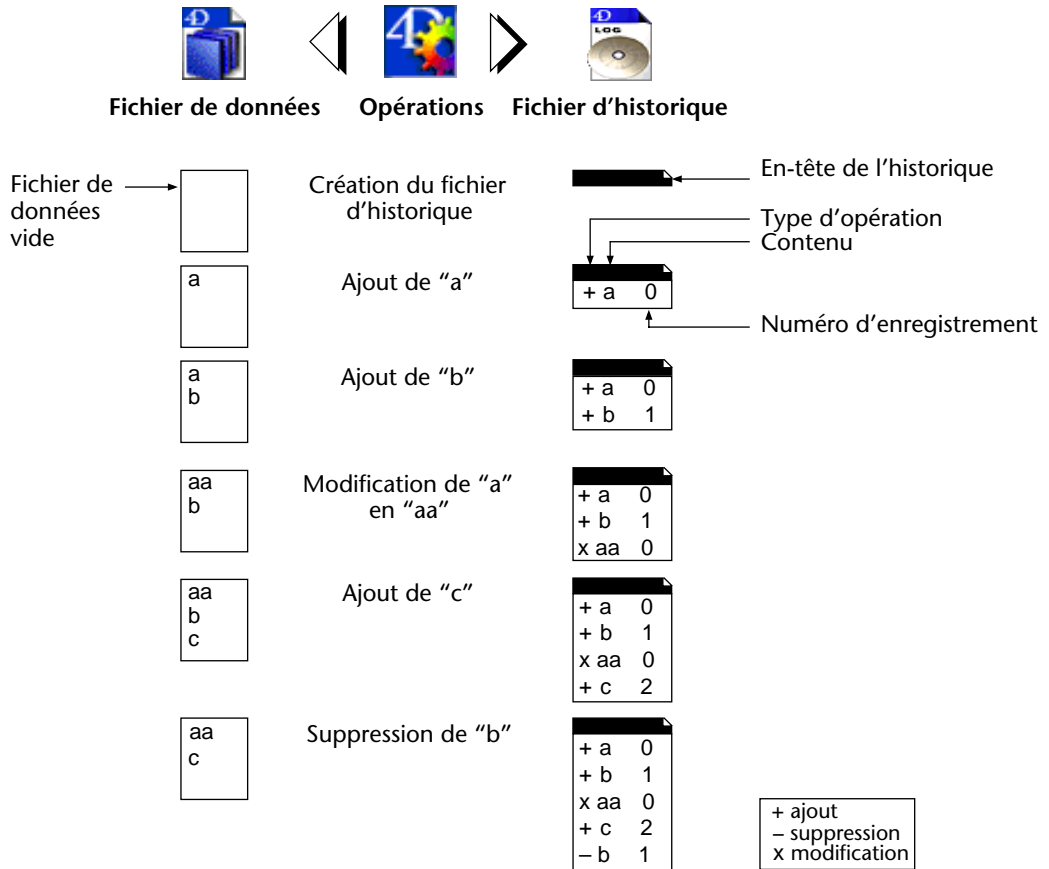
---

*Note sur les transactions* Une transaction est un ensemble d'opérations exécutées en bloc. Par exemple, lors d'un virement bancaire effectué dans une transaction, il faut être sûr de bien débiter le premier compte et de créditer le second dans la même opération, sans courir le risque d'interrompre l'opération en cours. Soit toute l'opération s'effectue, soit rien n'est effectué, mais en aucun cas elle ne peut s'effectuer à moitié. Il ne faut pas que le premier compte soit débité sans que le second ne soit crédité. Les transactions sont considérées par 4<sup>e</sup> Dimension comme une seule opération, au même titre que l'ajout, la modification ou la suppression.

---

Le fichier d'historique est géré par 4<sup>e</sup> Dimension. Il prend en compte indifféremment toutes les opérations affectant le fichier de données, que ce soient des manipulations effectuées par un utilisateur (à partir de 4<sup>e</sup> Dimension monoposte ou de 4D Client), des méthodes 4D, les plug-ins 4D (4D Write, 4D View, 4D Draw, etc.), un navigateur Web ou encore 4D Open dans le cas de 4D Server.

Ce schéma résume le principe général de fonctionnement du fichier d'historique :



Le fichier d'historique courant est automatiquement sauvegardé avec le fichier de données courant. Ce mécanisme procure deux avantages principaux :

- Eviter la saturation du disque accueillant le fichier d'historique. En effet, sans sauvegarde, l'historique grossirait indéfiniment au fur et à mesure de l'exploitation de la base et finirait par saturer votre disque. A

chaque sauvegarde du fichier de données, 4<sup>e</sup> Dimension ou 4D Server ferme le fichier d'historique courant et débute immédiatement un nouveau fichier vide, évitant ainsi le risque de saturation. L'ancien fichier d'historique est alors archivé puis éventuellement détruit, conformément au mécanisme des jeux de sauvegarde.

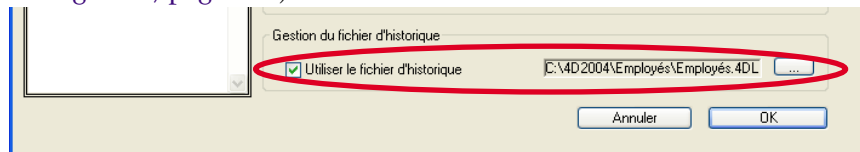
- Conserver les fichiers d'historique correspondant aux sauvegardes, afin de pouvoir analyser ou réparer *a posteriori* une base. En effet, l'intégration du fichier d'historique ne peut se faire que dans la base qui lui correspond. Il est donc important, pour pouvoir intégrer correctement un fichier d'historique dans une sauvegarde, de disposer de sauvegardes et d'historiques archivés simultanément.

## Créer le fichier d'historique

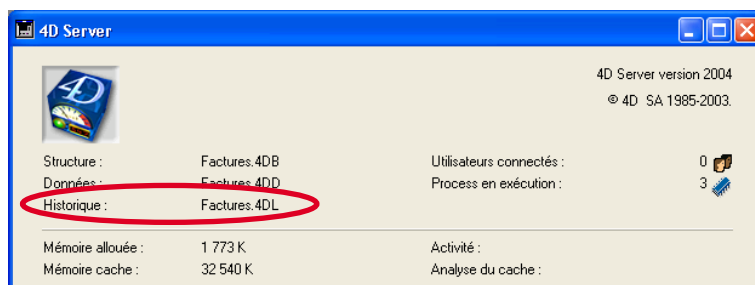
Par défaut, toute base créée avec 4<sup>e</sup> Dimension 2004 utilise un fichier d'historique : la création de ce fichier est en effet associée à l'option **Sauvegarde automatique** de la boîte de dialogue de création de base (cf. [paragraphe "Sauvegarde automatique", page 216](#)).

Le fichier d'historique est nommé **NomFichierDonnées.4DL** et est placé dans le dossier contenant la structure de la base.

Vous pouvez à tout moment savoir si votre base utilise un fichier d'historique : l'option **Utiliser le fichier d'historique** est cochée dans la page **Configuration** (cf. [paragraphe "Configuration des fichiers de sauvegarde", page 203](#)).



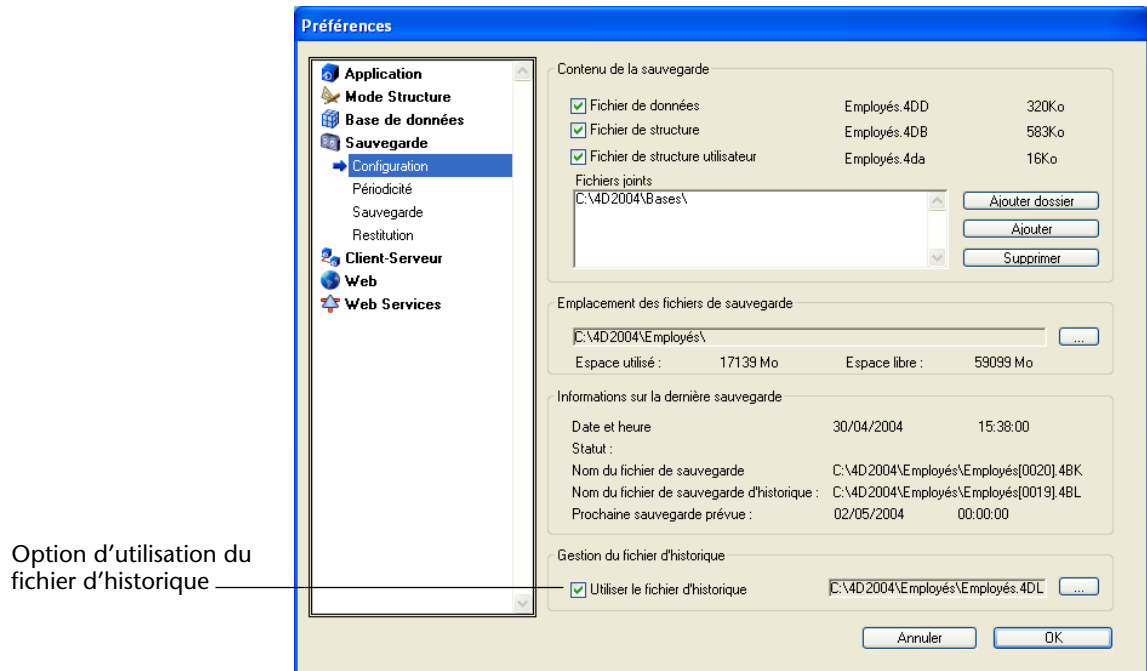
Avec 4D Server, le nom du fichier d'historique s'inscrit également dans la fenêtre du poste serveur :



Si vous avez désélectionné cette option ou si vous utilisez une base sans fichier d'historique et souhaitez mettre en place une stratégie de sauvegarde avec fichier d'historique, il vous appartient d'en créer un.

► Pour créer un fichier d'historique :

1 Dans la page **Configuration** (thème "Sauvegarde") des **Préférences de l'application**, cochez l'option **Utiliser le fichier d'historique**.



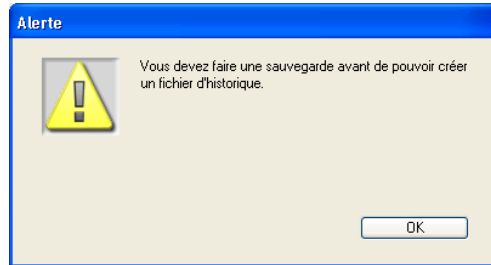
*Note de compatibilité* Cette option remplace l'option "Historique obligatoire" présente dans les versions précédentes de 4<sup>e</sup> Dimension.

Pour que vous puissiez créer un fichier d'historique, la base de données doit se trouver dans une des situations suivantes :

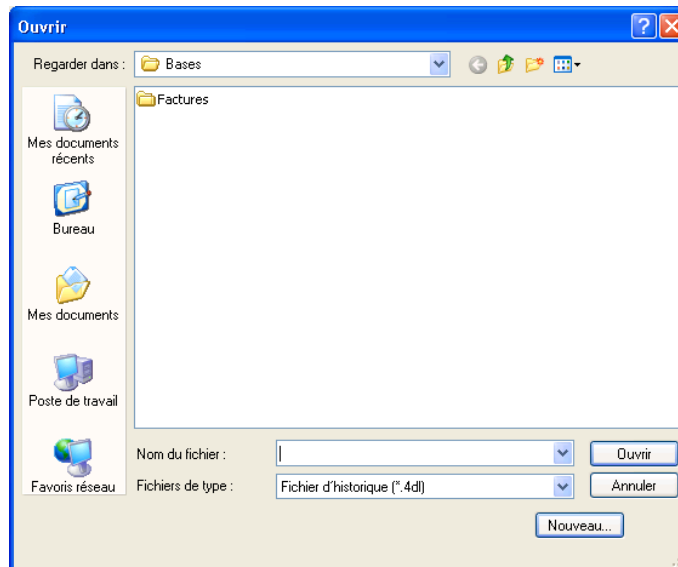
- le fichier de données est vierge,
- vous venez d'effectuer une sauvegarde de la base et aucune modification sur les données n'a encore été effectuée.

Dans tous les autres cas, le programme affiche une boîte de dialogue d'alerte vous demandant de réaliser une sauvegarde. Cette précaution est nécessaire car il vous faudra, pour restituer une base de données après un éventuel incident, disposer d'une copie de la base dans

laquelle pourront s'intégrer les opérations enregistrées dans le fichier d'historique.



Si la base se trouve dans une des conditions définies ci-dessus, le programme affiche une boîte de dialogue standard d'ouverture ou de création du fichier d'historique :



**2 Cliquez sur le bouton Nouveau...**

Une boîte de dialogue standard d'enregistrement de fichiers apparaît. Par défaut, le fichier d'historique est baptisé *NomFichierDonnées.4DL*.

**3 Conservez le nom du fichier par défaut ou choisissez-en un autre et déterminez l'emplacement du fichier.**

Si vous disposez d'au moins deux disques durs, il est recommandé de placer l'historique sur un autre disque que celui contenant la base, afin de pouvoir l'exploiter en cas de perte totale du disque de la base.

**4 Cliquez sur le bouton Enregistrer.**

Le chemin d'accès et le nom du fichier d'historique ouvert s'affiche alors dans la zone "Utiliser le fichier d'historique" de la boîte de dialogue des Préférences.

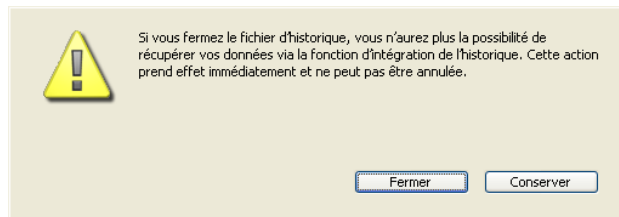
Sans autre manipulation de votre part, toutes les opérations effectuées sur les données s'inscriront dans ce fichier, et il sera utilisé également lors des ouvertures ultérieures de la base.

- Vous devrez créer un autre fichier d'historique si vous créez un nouveau fichier de données.
- Vous devrez désigner ou créer un autre fichier d'historique si vous ouvrez un autre fichier de données non associé à un fichier d'historique (ou si le fichier d'historique est manquant). Pour ouvrir un autre fichier d'historique, cliquez sur le bouton [...] situé à droite de la zone d'affichage du chemin d'accès du fichier d'historique.

## Fermer l'historique

Si vous souhaitez interrompre l'enregistrement du fichier d'historique courant, désélectionnez l'option **Utiliser le fichier d'historique** dans la page **Configuration** (thème "Sauvegarde") des Préférences de l'application.

4<sup>e</sup> Dimension affiche alors un message d'alerte afin d'attirer votre attention sur le fait que cette action vous prive de la sécurité apportée par le fichier d'historique :



Si vous cliquez sur le bouton **Fermer**, le fichier d'historique courant est immédiatement refermé (il n'est pas nécessaire que la boîte de dialogue des Préférences soit ensuite validée).

Si vous souhaitez fermer votre fichier d'historique courant parce qu'il devient trop important, il est préférable d'effectuer une sauvegarde du fichier de données, ce qui entraînera la sauvegarde de l'historique.

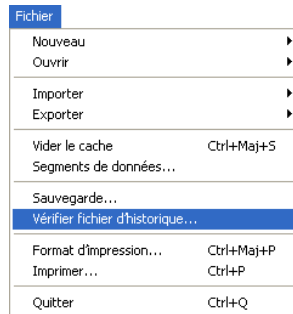
## Analyser l'historique

4<sup>e</sup> Dimension inclut une fonction permettant de visualiser le contenu du fichier d'historique courant. Cette fonction est utile pour analyser l'utilisation d'une base de données ou détecter la ou les opération(s) à

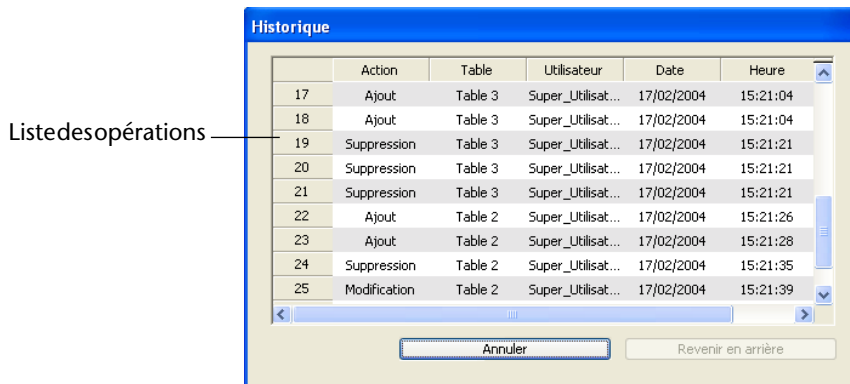
l'origine d'erreurs ou de dysfonctionnements. Dans le cadre d'une base de données exploitée en client-serveur, elle permet de vérifier les opérations effectuées par chaque poste client.

Cette fonction permet également de revenir en arrière parmi les opérations effectuées sur les données de la base. Cette possibilité est traitée dans le [paragraphe "Remonter en arrière parmi les opérations"](#), page 234.

Pour visualiser l'historique courant, sélectionnez la commande **Vérifier fichier d'historique...** dans le menu **Fichier** de 4<sup>e</sup> Dimension ou de 4D Client (cette fonction n'est pas accessible sur le poste serveur) :



La fenêtre suivante apparaît :



Chaque opération élémentaire enregistrée dans le fichier d'historique apparaît sous forme d'une ligne. Les colonnes fournissent diverses informations concernant l'opération. Vous pouvez modifier l'affichage par défaut des colonnes — et notamment afficher les valeurs de champs — à l'aide du menu contextuel de la fenêtre (cf. [paragraphe "Modifier l'affichage des colonnes par défaut"](#), page 225).



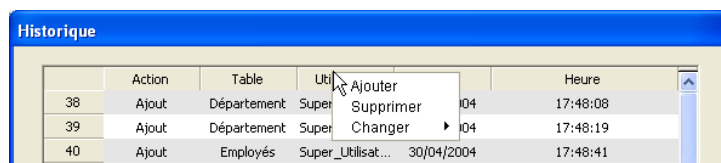
Les informations affichées par défaut permettent d'identifier la source et le contexte de chaque opération :

- **Action** : type d'opération effectuée sur les données. Cette colonne peut contenir une des quatre opérations suivantes : Ajout, Suppression, Modification et Transaction. Pour plus d'informations, reportez-vous au [paragraphe "Présentation", page 218](#).
- **Table** : table à laquelle appartient l'enregistrement ajouté/supprimé/modifié.
- **Utilisateur** : nom de l'utilisateur ayant effectué l'opération ; en version client-serveur, il s'agit du nom de la machine du poste client. En version monoposte, il s'agit du nom d'identification de l'utilisateur. Si les mots de passe de 4<sup>e</sup> Dimension ne sont pas activés, cette colonne contient Super\_Utilisateur. Si l'opération a été effectuée via 4D Open, le libellé "4D Open" est affiché.
- **Date et Heure** : date et heure à laquelle l'opération a été effectuée.

### Modifier l'affichage des colonnes par défaut

Vous pouvez modifier l'affichage des colonnes à l'aide du menu contextuel de la fenêtre d'historique. Vous pouvez ajouter, supprimer ou remplacer le contenu de chaque colonne. Les modifications effectuées dans la fenêtre sont conservées tant qu'elle n'est pas refermée.

Pour afficher le menu contextuel de la fenêtre, cliquez sur un en-tête de colonne :



Les commandes suivantes sont disponibles :

- **Ajouter** : permet d'ajouter une colonne à droite des colonnes de la fenêtre. Par défaut, la colonne est vide et libellée "Champ". Vous pourrez par la suite lui affecter un champ de la base (cf ci-dessous).
- **Supprimer** : permet d'enlever la colonne courante de la fenêtre.

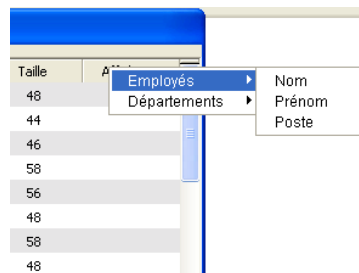
- **Remplacer** : permet de remplacer le contenu de la colonne courante. Vous pouvez sélectionner soit un des autres types d'information standard (Action, Table, etc.), soit **Champ** afin d'afficher une colonne à laquelle vous pourrez affecter un champ de la base.

**Visualiser les valeurs des champs**

Vous pouvez affecter des champs aux colonnes de la fenêtre d'analyse de l'historique afin d'afficher leur valeur courante lors de chaque opération d'ajout ou de modification. Cette possibilité est particulièrement utile par exemple pour identifier l'instant où un enregistrement spécifique a été ajouté.

Pour cela, il vous suffit d'ajouter une colonne "Champ" vide dans la fenêtre à l'aide de la commande **Ajouter** ou **Remplacer** du menu contextuel de la fenêtre (cf. paragraphe précédent).

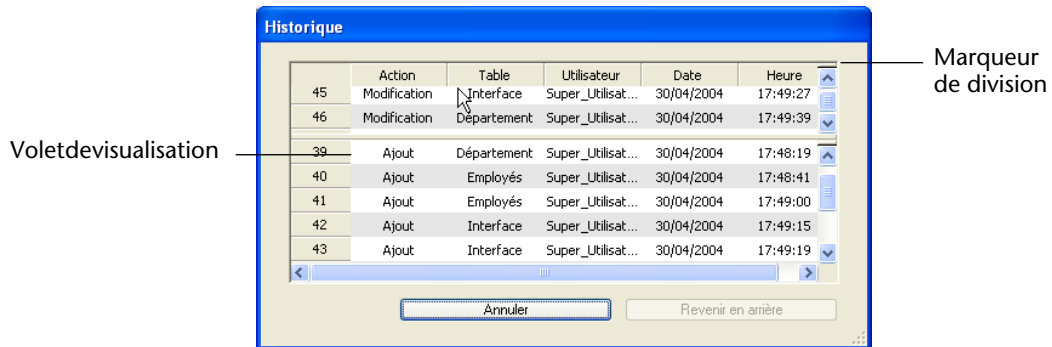
Cliquez à nouveau sur l'en-tête de la colonne Champ : la liste des tables et des champs de la base est alors affichée sous forme de menu hiérarchique :



Sélectionnez le champ dont vous souhaitez afficher les valeurs. Elles apparaissent alors pour chaque opération d'ajout ou de modification :

Historique						
	Action	Table	Utilisateur	Date	Heure	Champ
38	Ajout	Département	Super_Utilisat...	30/04/2004	17:48:08	
39	Ajout	Département	Super_Utilisat...	30/04/2004	17:48:19	
40	Ajout	Employés	Super_Utilisat...	30/04/2004	17:48:41	Valet
41	Ajout	Employés	Super_Utilisat...	30/04/2004	17:49:00	Frances
42	Ajout	Interface	Super_Utilisat...	30/04/2004	17:49:15	Burnet
43	Ajout	Interface	Super_Utilisat...	30/04/2004	17:49:19	
44	Ajout	Interface	Super_Utilisat...	30/04/2004	17:49:24	
45	Modification	Interface	Super_Utilisat...	30/04/2004	17:49:27	
46	Modification	Département	Super_Utilisat...	30/04/2004	17:49:39	

**Volets de visualisation** Vous pouvez créer des volets de visualisation dans la fenêtre d'historique. Pour cela, cliquez sur le marqueur de division situé en haut de la fenêtre et faites-le glisser vers le bas.



Pour créer plusieurs volets, il suffit de répéter l'opération autant de fois que nécessaire. Vous pouvez créer un volet au-dessous d'un volet existant en faisant glisser son marqueur de division.

Pour supprimer un volet, faites glisser sa ligne de séparation inférieure vers le haut de la fenêtre ou double-cliquez sur cette ligne de séparation.

## Récupération des données

Le module de sauvegarde intégré de 4<sup>e</sup> Dimension vous permet de récupérer l'intégralité des données d'une base en cas d'incident, quelles que soient les causes de l'incident.

### Incidents et diagnostic

Deux catégories principales d'incidents peuvent se produire :

- **L'arrêt inopiné de la base pendant son exploitation.**

Cet incident peut se produire à cause d'une coupure de courant, la panne d'un élément du système, etc. Dans ce cas, en fonction de l'état courant du cache de données au moment de l'incident, la récupération de la base peut nécessiter différentes opérations :

- si le cache était vide, la base s'ouvre normalement. Toutes les modifications apportées à la base ont été enregistrées. Ce cas ne nécessite aucune opération particulière.
- si le cache contenait des opérations, le fichier de données est intact mais il est nécessaire d'intégrer le fichier d'historique courant.

- si le cache était en cours d'écriture, le fichier de données est probablement endommagé. Il est nécessaire de restituer la dernière sauvegarde et d'intégrer le fichier d'historique courant.
- **La perte d'un ou plusieurs fichier(s) de la base.**  
Cet incident peut être causé par des secteurs défectueux sur le disque contenant la base, un virus, une erreur de manipulation, etc.  
Il est nécessaire de restituer la dernière sauvegarde puis d'intégrer éventuellement l'historique courant.

Pour savoir si une base a été endommagée à la suite d'un incident, il suffit de la relancer avec 4<sup>e</sup> Dimension. Le programme effectue un auto-diagnostic et précise les opérations de réparation à effectuer. En mode automatique, ces opérations sont effectuées directement, sans intervention de l'utilisateur (cf. [paragraphe "Réparations automatiques"](#), page 228).

Si une stratégie de sauvegarde régulière a été mise en place, les outils de récupération de 4<sup>e</sup> Dimension vous permettront dans la plupart des cas de retrouver la base dans l'état exact où elle se trouvait avant l'incident. Vous disposez de deux outils principaux :

- la restitution des sauvegardes,
- l'intégration du fichier d'historique.

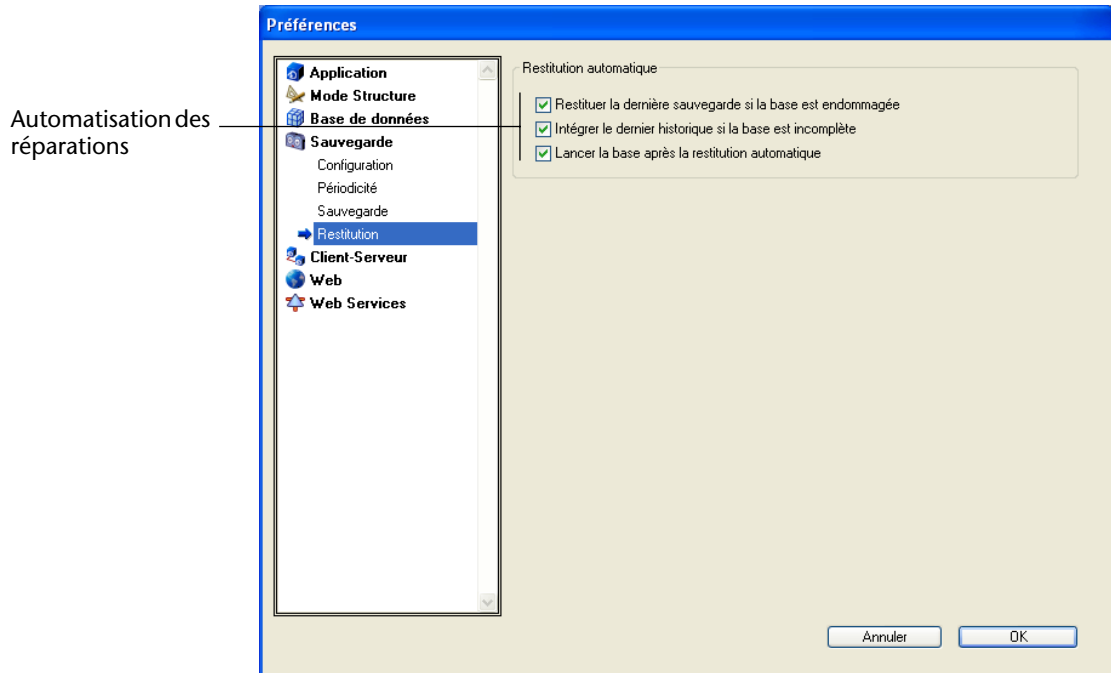
Ces deux fonctions peuvent être combinées et automatisées.

### Réparations automatiques

Par défaut, 4<sup>e</sup> Dimension lance automatiquement les procédures de récupération des bases après incident. Trois types d'automatismes sont utilisés :

- automatisation de la restitution
- automatisation de l'intégration de l'historique
- automatisation du démarrage d'une base restituée.

Ces mécanismes peuvent être désactivés à l'aide d'options accessibles dans la page **Restitution** (thème "Sauvegarde") de la fenêtre des Préférences de l'application :



### Restituer la dernière sauvegarde si la base est endommagée

Lorsque cette option est cochée, le programme déclenche automatiquement la restitution du fichier de données de la dernière sauvegarde valide de la base s'il détecte une anomalie (fichier corrompu par exemple) lors du lancement de la base. Aucune intervention de l'utilisateur n'est requise ; l'opération est cependant consignée dans le journal des sauvegardes (cf. [paragraphe "Le journal des sauvegardes"](#), page 236).

- Notes*
- La restitution d'une sauvegarde consiste à extraire les fichiers contenus au sein d'un ou de plusieurs fichier(s) de sauvegarde et à les rendre à nouveau lisibles pour l'application qui les a créés.
  - En cas de restitution automatique, seul le fichier de données est restitué. Si vous souhaitez récupérer les fichiers joints ou le fichier de structure, vous devez effectuer une restitution manuelle.

Les étapes de la restitution automatique sont les suivantes :

- Dans un premier temps, 4<sup>e</sup> Dimension renomme le fichier de données endommagé.
- 4<sup>e</sup> Dimension extrait le fichier de données de la dernière sauvegarde et le stocke à la place du précédent.
- Si l'option d'intégration automatique de l'historique est cochée, 4<sup>e</sup> Dimension intègre si nécessaire le fichier d'historique (cf. [paragraphe "Intégrer le dernier historique si la base est incomplète", page 230](#)).
- Si l'option de démarrage automatique de la base restituée est cochée, 4<sup>e</sup> Dimension relance la base restituée. (cf. [paragraphe "Lancer la base après la restitution automatique", page 230](#)).

### **Intégrer le dernier historique si la base est incomplète**

Lorsque cette option est cochée, le programme intègre automatiquement l'historique lors de l'ouverture ou de la restitution d'une base de données. Aucune intervention de l'utilisateur n'est requise ;

- Lors de l'ouverture d'une base, l'historique courant est automatiquement intégré si 4<sup>e</sup> Dimension détecte que des opérations stockées dans l'historique ne sont pas présentes dans les données. Cette situation se produit par exemple lorsqu'une panne de courant a lieu alors que des opérations non encore écrites sur le disque se trouvaient dans le cache de données.
- Lors de la restitution d'une base, si le fichier d'historique courant ou un fichier de sauvegarde d'historique ayant le même numéro que le fichier de sauvegarde est stocké dans le même dossier, 4<sup>e</sup> Dimension examine son contenu. S'il contient des opérations non présentes dans le fichier de données, le programme l'intègre automatiquement.

Aucune boîte de dialogue n'est présentée à l'utilisateur, l'opération est entièrement automatique. Le but est de faciliter au maximum la remise en route de l'exploitation. L'opération est consignée dans le journal des sauvegardes (cf. [paragraphe "Le journal des sauvegardes", page 236](#)).

### **Lancer la base après la restitution automatique**

Lorsque cette option est cochée, 4<sup>e</sup> Dimension ouvre automatiquement une base venant d'être restituée, à l'issue de la procédure de restitution.

A noter que lorsque cette option est utilisée, il est important que tous les éléments nécessaires au fonctionnement de la base soient accessibles après la restitution. Si, par exemple, la base requiert des fichiers spécifiques à côté du fichier de structure de la base, ils doivent être inclus dans les sauvegardes afin d'être restitués en même temps que la base. Si la base utilise des plug-ins, il est préférable qu'ils soient installés à côté de l'application 4<sup>e</sup> Dimension.

Le cumul des trois options de réparation (associées aux options de sauvegarde) permet de mettre en oeuvre un système de remise en route semi-automatique ou automatique après incident.

- Remise en route semi-automatique : il suffit à l'administrateur de relancer la base, l'application 4<sup>e</sup> Dimension effectue automatiquement les réparations nécessaires et relance une version fiable et à jour de l'application.
- Remise en route automatique (4D Server uniquement) : si vous enregistrez la base comme service exécuté au démarrage du système (cf. [paragraphe "Enregistrer 4D Server comme service sous MacOS X", page 26](#)), elle sera automatiquement relancée par la machine au redémarrage. L'application 4<sup>e</sup> Dimension effectue automatiquement les réparations nécessaires.

Ce type de système est particulièrement adapté aux bases de données 4D Server. A noter que chaque opération exécutée automatiquement est consignée dans le journal des sauvegardes de la base (cf. [paragraphe "Le journal des sauvegardes", page 236](#)).

## Restitution manuelle d'une sauvegarde

Vous pouvez restituer manuellement le contenu d'une archive générée par le module de sauvegarde. Une restitution manuelle peut être nécessaire par exemple pour restituer la totalité du contenu d'une archive (fichiers de structure et/ou fichiers joints inclus) ou à des fins de recherche sur des archives.

La restitution manuelle peut éventuellement s'accompagner de l'intégration de l'historique courant.

---

*Note* Vous pouvez paramétrer 4<sup>e</sup> Dimension de manière à ce que les données de la base soient automatiquement restituées en cas d'incident. Pour plus d'informations sur ce point, reportez-vous au [paragraphe "Réparations automatiques", page 228](#).

---

La restitution des sauvegardes est réalisée par l'application 4D elle-même (4<sup>e</sup> Dimension, 4D Server, application fusionnée avec 4D Runtime Volume License, etc.), via la page **Restitution** de la boîte de dialogue de bienvenue.

► Pour restituer manuellement une sauvegarde :

1 Dans la boîte de dialogue de bienvenue, cliquez sur l'onglet **Restituer** (4<sup>e</sup> Dimension et 4D Server uniquement).

**OU BIEN**

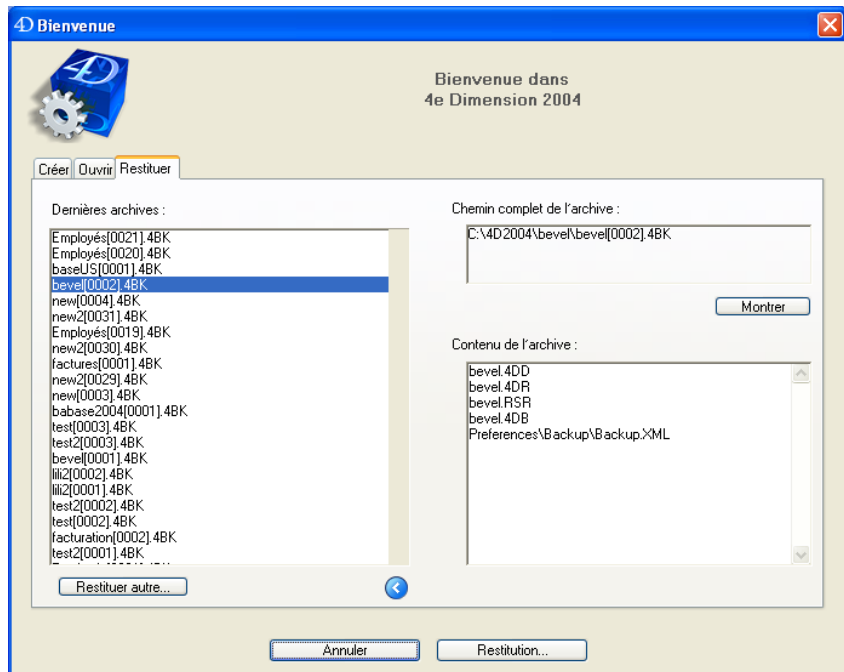
Exécutez la commande **RESTITUER** depuis une méthode de 4<sup>e</sup> Dimension (toutes applications 4D).

**OU BIEN**

Faites glisser et déposez le fichier de sauvegarde à restituer sur l'icône de l'application (toutes applications 4D).

Dans ce dernier cas, une boîte de dialogue standard d'enregistrement de fichiers apparaît, vous permettant d'indiquer l'emplacement auquel vous souhaitez que les fichiers soient restitués. Passez directement à l'étape 3.

2 Dans la zone "Dernières archives", sélectionnez la base puis la sauvegarde à restituer et cliquez sur le bouton **Restituer...**



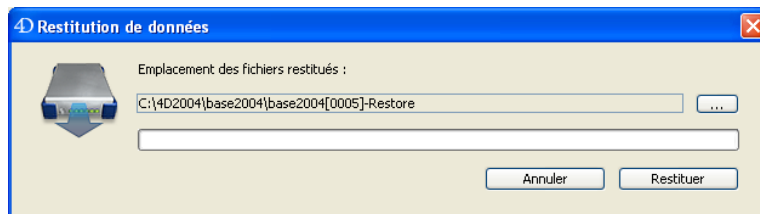


La zone d'option de la boîte de dialogue indique le chemin d'accès et le contenu de l'archive sélectionnée. Vous pouvez visualiser l'emplacement du fichier de sauvegarde sur le disque en cliquant sur le bouton **Montrer**. Si la sauvegarde a été segmentée, le chemin d'accès du premier segment est affiché.

**OU BIEN :**

**Pour restituer une autre archive, cliquez sur le bouton Restituer autre... puis désignez le fichier de sauvegarde à restituer dans la boîte de dialogue standard d'ouverture de document.**

Une boîte de dialogue apparaît, vous permettant de désigner l'emplacement auquel vous souhaitez que les fichiers soient restitués :



Par défaut, 4<sup>e</sup> Dimension restitue les fichiers dans un dossier nommé "Nomarchive-Restore" placé à côté de l'archive. Vous pouvez cliquer sur le bouton [...] et indiquer un autre emplacement.

### 3 Modifiez si nécessaire l'emplacement des fichiers restitués et cliquez sur le bouton Restituer.

4<sup>e</sup> Dimension extrait tous les fichiers de la sauvegarde à l'emplacement défini.

Si le fichier d'historique courant ou un fichier de sauvegarde d'historique ayant le même numéro que le fichier de sauvegarde est stocké dans le même dossier, 4<sup>e</sup> Dimension examine son contenu. S'il contient des opérations non présentes dans le fichier de données, le programme propose de l'intégrer.

*Note* L'intégration est effectuée automatiquement si l'option d'intégration automatique de l'historique est cochée (cf. [paragraphe "Intégrer le dernier historique si la base est incomplète", page 230](#)).

### 4 (Facultatif) Cliquez sur OK pour intégrer le fichier d'historique dans la base restituée.

Si les opérations de restitution et d'intégration se sont déroulées correctement, 4<sup>e</sup> Dimension affiche une boîte de dialogue proposant d'ouvrir la base.

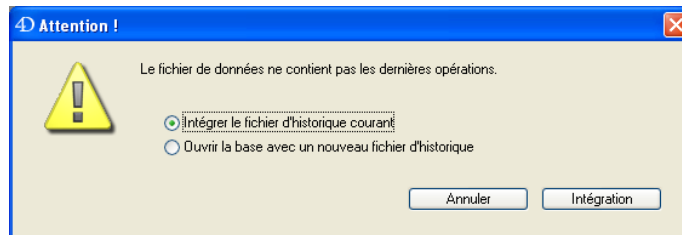
- 5 Cliquez sur **OK** pour lancer la base de données restituée.  
OU  
Cliquez sur **Annuler** pour quitter l'application 4<sup>e</sup> Dimension.

*Note* La base est automatiquement lancée si l'option de redémarrage automatique est cochée (cf. [paragraphe "Lancer la base après la restitution automatique", page 230](#)).

Lors de la restitution, 4<sup>e</sup> Dimension place tous les fichiers sauvegardés dans le dossier d'arrivée, quelle que soit la position des fichiers originaux sur le disque au moment de la sauvegarde. De cette façon, vous retrouverez plus facilement vos fichiers.

## Intégration manuelle de l'historique

Si vous n'avez pas coché l'option d'intégration automatique du fichier d'historique (cf. [paragraphe "Intégrer le dernier historique si la base est incomplète", page 230](#)), une boîte de dialogue d'alerte apparaît à l'ouverture de la base lorsque 4<sup>e</sup> Dimension constate que le fichier d'historique contient plus d'opérations qu'il n'en a été effectué dans la base.



Pour que ce mécanisme fonctionne, 4<sup>e</sup> Dimension doit être en mesure d'accéder au fichier d'historique à son emplacement courant.

Vous pouvez choisir d'intégrer ou non les données de l'historique courant. Ne pas intégrer l'historique courant permet notamment d'éviter de reproduire des erreurs effectuées sur les données.

## Remonter en arrière parmi les opérations

4<sup>e</sup> Dimension dispose d'une fonction permettant de remonter en arrière parmi les opérations effectuées sur le fichier d'historique. Cette fonction s'apparente à une fonction d'annulation sur plusieurs niveaux. Elle est utile notamment lorsqu'un enregistrement a été supprimé par erreur dans une base de données.

Pour que cette fonction soit accessible, il est impératif que la base travaille avec un fichier d'historique.

► Pour restituer la base dans un état précédent :

- 1 **Sélectionnez la commande Vérifier fichier d'historique... dans le menu Fichier de 4<sup>e</sup> Dimension ou de 4D Client (cette fonction n'est pas accessible sur le poste serveur).**

La fenêtre de visualisation du fichier d'historique apparaît. Cette fenêtre est décrite dans le [paragraphe "Analyser l'historique"](#), page 223.

- 2 **Sélectionnez la ligne après laquelle toutes les opérations doivent être annulées.**

L'opération de la ligne sélectionnée sera la dernière conservée. Si par exemple vous souhaitez annuler une suppression, sélectionnez l'opération située juste avant la suppression. L'opération de suppression ainsi que toutes les opérations suivantes seront annulées :

Opération sélectionnée

	Action	Table	Utilisateur	Date	Heure
47	Ajout	Interface	Super_Utilisat...	30/04/2004	18:58:48
48	Ajout	Interface	Super_Utilisat...	30/04/2004	18:58:48
49	Ajout	Interface	Super_Utilisat...	30/04/2004	18:58:48
50	Suppression	Interface	Super_Utilisat...	30/04/2004	18:58:56
51	Suppression	Interface	Super_Utilisat...	30/04/2004	18:58:56
52	Suppression	Interface	Super_Utilisat...	30/04/2004	18:58:56
53	Suppression	Interface	Super_Utilisat...	30/04/2004	18:58:56
54	Ajout	Interface	Super_Utilisat...	30/04/2004	18:58:58
55	Ajout	Interface	Super Utilisat...	30/04/2004	18:58:58

Opérations conservées

Opérations annulées

Annuler Revenir en arrière

- 3 **Cliquez sur le bouton Revenir en arrière.**

4<sup>e</sup> Dimension vous demande de confirmer l'opération.

- 4 **Cliquez sur OK.**

Les données sont alors restituées dans l'état exact où elles se trouvaient au moment de l'action sélectionnée.

### Principe de fonctionnement

Le principe de fonctionnement de cette option est le suivant : lorsque l'utilisateur clique sur le bouton **Revenir en arrière**, 4<sup>e</sup> Dimension ferme la base de données courante et restitue la dernière sauvegarde des données de la base. La base restituée est ensuite ouverte et 4<sup>e</sup> Dimension intègre les opérations de l'historique jusqu'à l'opération sélectionnée.

Si la base n'avait pas encore été sauvegardée, 4<sup>e</sup> Dimension repart d'un fichier de données vierge.

## Le journal des sauvegardes

Pour faciliter le suivi et la vérification de la sauvegarde de vos bases, le module de sauvegarde écrit un récapitulatif de chaque opération qu'il effectue dans un fichier particulier, qui est en quelque sorte le journal de son activité. A la façon d'un livre de bord, toutes les opérations concernant une base (sauvegardes, restitutions, intégrations d'historiques) sont consignées dans ce fichier, qu'elles aient eu lieu automatiquement ou manuellement. Ces opérations s'accompagnent de la date et l'heure à laquelle elles ont eu lieu.

Dans le cas d'une sauvegarde, le journal consigne les informations suivantes :

Pour chaque archive :

- Chemin d'accès (pour l'archive ou l'historique créé(e) ou supprimé(e))

- Taille

- Contenu de l'archive (fichier de données, fichier de structure, fichier de structure utilisateur, fichiers annexes, fichier d'historique)

Pour chaque élément contenu dans l'archive :

- Type du fichier

- Taille du fichier original

- Date/Heure (effective de début, prévue de début, effective de fin)

Le journal des sauvegardes est nommé "Journal des sauvegardes.txt" et est placé à côté du fichier de données de la base.

Le journal des sauvegardes peut être ouvert par n'importe quel éditeur de texte. Afin de faciliter son analyse, les informations qu'il contient sont séparées par des tabulations et chaque ligne d'informations se termine par un retour chariot.

## Fichier XML des paramètres

La totalité des préférences de sauvegarde et de restitution de 4<sup>e</sup> Dimension sont disponibles sous forme de fichier XML indépendant. 4<sup>e</sup> Dimension exploite les données de ce fichier pour afficher les options dans la boîte de dialogue des Préférences et au moment du déclenchement de chaque sauvegarde.

Il est donc possible de lire et de modifier les paramètres de sauvegarde à l'aide des commandes XML de 4<sup>e</sup> Dimension (cf. [paragraphe "XML", page 374](#)) ou via tout éditeur XML. Ce principe permet aux

développeurs de gérer les paramètres de sauvegarde par programmation, notamment dans des applications compilées et fusionnées avec 4D Runtime Volume License :

Affichage du fichier XML des paramètres de sauvegarde dans un navigateur

```

D:\4D2004\Factures\Preferences\Backup\Backup.XML - Microsoft Internet Explorer
Fichier  Edition  Affichage  Favoris  Outils  ?
Précédente  Recherche  Favoris  Média
Adresse  D:\4D2004\Factures\Preferences\Backup\Backup.XML  OK  Liens

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <Preferences4D>
- <Backup>
- <DataBase>
- <DatabaseName>
  <ItemsCount>1</ItemsCount>
  <Item1>D:\4D2004\Factures\Factures.4DD</Item1>
</DatabaseName>
+ <LastBackupPath>
- <LastBackupLogPath>
  <ItemsCount>1</ItemsCount>
  <Item1 />
</LastBackupLogPath>
- <CurrentBackupSet>
  <ItemsCount>1</ItemsCount>
  <Item1>0</Item1>
</CurrentBackupSet>
- <LastBackupDate>
  <ItemsCount>1</ItemsCount>
  <Item1>-29-01-0000</Item1>
</LastBackupDate>
- <LastBackupTime>
  <ItemsCount>1</ItemsCount>
  <Item1>00:00:-1+02:00</Item1>

```

Ce fichier contient deux ensembles de préférences :

- les **paramètres généraux** paramètres de sauvegarde généraux et avancés de la sauvegarde.
- les **paramètres spécifiques au fichier de données** paramètres relatifs à la dernière sauvegarde (date et heure de la dernière sauvegarde, date et heure de la prochaine sauvegarde, emplacement du dernier fichier de sauvegarde, numéro de la dernière sauvegarde).

Le fichier XML des préférences de sauvegarde et de restitution est nommé **Backup.XML**. Il est créé automatiquement à l'emplacement suivant :

**DossierBase/Preferences/Backup/Backup.XML**

... où **DossierBase** représente le dossier contenant le fichier de structure de la base. Les sous-dossiers **Preferences/Backup/** sont créés automatiquement si nécessaire.

## A propos de 4D Backup

Ce paragraphe s'adresse aux utilisateurs de 4D Backup, l'ancien plug-in de sauvegarde et de restitution de 4<sup>e</sup> Dimension. Il établit une comparaison entre les fonctions du nouveau module de sauvegarde de 4<sup>e</sup> Dimension 2004 et celles de 4D Backup.

---

*Note* 4D Backup n'est pas compatible avec 4<sup>e</sup> Dimension version 2004 et suivantes.

---

### Compatibilité

Les algorithmes et les mécanismes utilisés par le module de sauvegarde de 4<sup>e</sup> Dimension 2004 sont entièrement nouveaux. Par conséquent, les projets de sauvegarde de 4D Backup ne sont pas compatibles avec le module de sauvegarde de 4<sup>e</sup> Dimension 2004.

De même, les archives générées par 4D Backup ne peuvent pas être restituées par le module de sauvegarde, et inversement.

En revanche, la structure du fichier d'historique est inchangée. Les fichiers d'historique générés avec des versions de 4<sup>e</sup> Dimension antérieures à la 2004 peuvent être exploités par cette nouvelle version.

### Fonctions de 4D Backup non maintenues

Ce paragraphe liste les fonctions de 4D Backup non disponibles dans le nouveau module de sauvegarde. Certaines de ces fonctions seront ajoutées dans les prochaines versions du module.

- Prise en charge spécifique des différents supports (disquettes, volumes éjectables, DAT, etc.).  
Le nouveau module de sauvegarde gère de la même manière chaque support affiché par le système d'exploitation.
- Segmentation automatique en cas d'espace disque insuffisant.  
Il vous appartient de vérifier que l'espace disque est suffisant pour accueillir les archives.
- Sauvegarde multi-volumes.  
Le nouveau module de sauvegarde permet cependant de segmenter les archives de manière à pouvoir les recopier sur plusieurs volumes.
- Miroir logique.  
Cette fonction n'est pas implémentée dans la première version du nouveau module de sauvegarde.

## Avantages du nouveau module

Ce paragraphe liste les principales fonctions du nouveau module de sauvegarde apportant un “plus” par rapport à 4D Backup.

- Simplification de la gestion des sauvegardes : stratégie de sauvegarde par défaut, intégration à 4<sup>e</sup> Dimension, interface harmonisée.
- Nomenclature des fichiers d’archives simplifiée et uniformisée entre les différentes plates-formes et les différentes langues.
- Programmation des sauvegardes et des options avec 4<sup>e</sup> Dimension, 4D Server ou les applications fusionnées avec 4D Runtime Volume License.
- Automatisation des sauvegardes et des restitutions avec 4<sup>e</sup> Dimension, 4D Server ou les applications fusionnées avec 4D Runtime Volume License.
- Fonction d’annulation multi-niveaux des opérations effectuées sur les données.
- Compression et sécurité accrue des archives (entrelacement, redondance).
- Accès programmé en XML à toutes les options de sauvegarde et de restitution.





# 6

## Serveur Web

Les possibilités offertes par le serveur Web de 4<sup>e</sup> Dimension 2004, 4D Server 2004 et 4D Client 2004 ont été étendues. Les nouveautés sont les suivantes :

- Prise en charge des connexions persistantes (*keep-alive*).
- Possibilité de modifier le port TCP utilisé pour les connexions sécurisées (HTTPS).
- Modification de la prise en charge des URLs inconnus.
- Nouvelle solution pour le lancement du serveur Web sous MacOS X.
- Nouvelle commande de menu pour tester le serveur Web.
- Ajout des nouvelles commandes [TRAITER BALISES HTML](#), [LIRE CORPS HTTP](#) et [ENVOYER DONNEES HTTP](#). Ces nouvelles commandes sont décrites dans le chapitre “Langage” de ce manuel, [paragraphe “Serveur Web”, page 360](#).
- Modification de la prise en charge des CGI — ajout d’une nouvelle commande spécifique. Cette nouveauté est décrite dans le chapitre “Langage”, [paragraphe “Prise en charge des CGI”, page 366](#).

---

*Note* Le serveur Web de 4D exploite désormais pleinement le protocole HTTP/1.1.

---

### Connexions persistantes

L’une des principales conséquences de la prise en charge du protocole de communication HTTP/1.1 est que le serveur Web de 4<sup>e</sup> Dimension peut désormais utiliser des connexions persistantes (*keep-alive*).

## Quel est l'intérêt des connexions persistantes ?

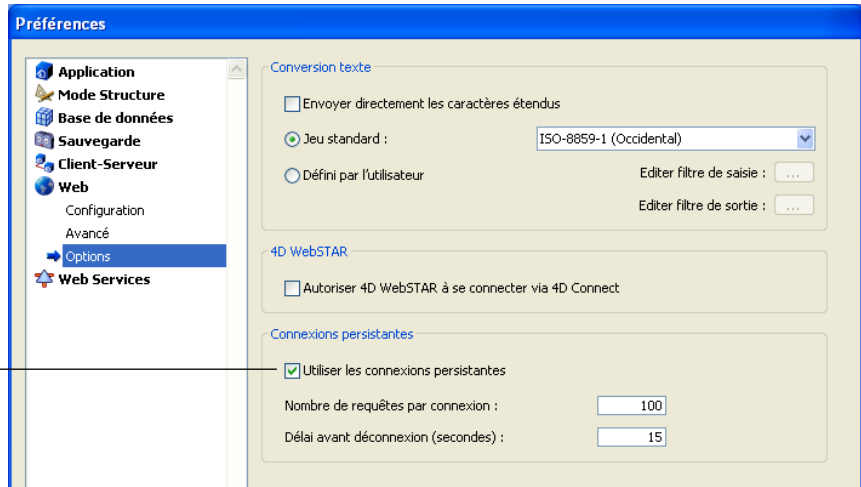
En HTTP/1.0, une connexion TCP est ouverte puis refermée par le serveur pour chaque requête. Si par exemple, une page HTML comporte 10 images, l'affichage de la page sur un navigateur provoque l'ouverture et la fermeture de 11 connexions TCP sur le serveur. Or, l'ouverture et la fermeture d'une connexion TCP mobilise une certaine quantité de temps CPU, de bande passante et de mémoire sur le serveur Web.

L'option *keep-alive* permet de maintenir ouverte une seule connexion TCP pour l'ensemble des échanges effectués par un navigateur et le serveur afin d'économiser les ressources et d'optimiser les échanges.

## Paramétrer les connexions persistantes

Une nouvelle option, présente dans la page **Options** des Préférences "Web" de 4<sup>e</sup> Dimension, active ou inactive les connexions TCP persistantes pour le serveur Web :

Gestion des connexions persistantes



Cette option est cochée par défaut. Dans la plupart des cas, il est conseillé de conserver cette option cochée car elle permet d'accélérer les échanges. Si le navigateur Web ne prend pas en charge les connexions *keep-alive*, le serveur Web 4D bascule automatiquement en HTTP/1.0.

La fonction *keep-alive* du serveur Web de 4<sup>e</sup> Dimension concerne toutes les connexions TCP/IP (HTTP, HTTPS), en mode contextuel et en mode sans contexte.

A noter toutefois que les connexions persistantes ne sont pas systématiquement utilisées pour tous les process Web 4D. Dans

certains cas, d'autres mécanismes d'optimisation du serveur Web sont mis en oeuvre. Les connexions persistantes sont principalement efficaces lors de l'envoi de pages statiques.

Les options **Nombre de requêtes par connexion** et **Délai avant déconnexion** permettent de régler le fonctionnement du mécanisme des connexions persistantes :

- **Nombre de requêtes par connexion** : permet de définir le nombre maximum de requêtes pouvant transiter dans une même connexion persistante. Limiter le nombre de requêtes par connexion permet d'éviter les risques de saturation du serveur via l'envoi massif de requêtes (technique utilisée par les pirates).  
La valeur par défaut (100) peut être réduite ou augmentée en fonction des ressources de la machine hébergeant le serveur Web 4D.
- **Délai avant déconnexion** (*timeout*) : cette valeur définit le délai maximal (en secondes) pendant lequel le serveur Web maintient ouverte une connexion TCP tant qu'aucune nouvelle requête n'est reçue de la part du navigateur. A l'issue de ce délai, le serveur referme la connexion.

Si le navigateur envoie une requête après la fermeture de la connexion, une nouvelle connexion TCP est automatiquement créée. Ce fonctionnement est transparent pour l'utilisateur.

## Modifier le numéro de port HTTPS

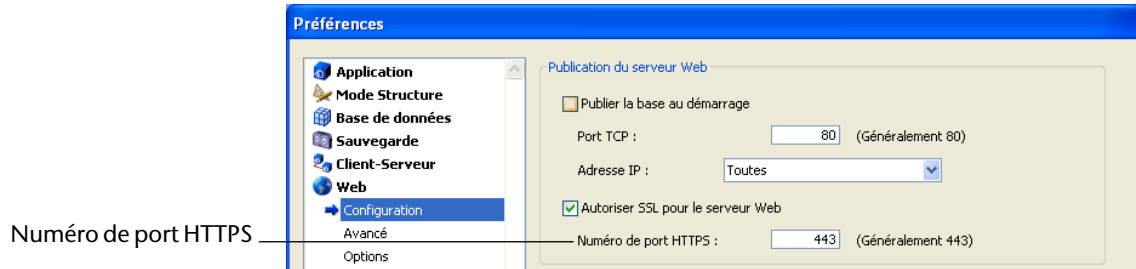
Vous pouvez désormais modifier le numéro du port TCP/IP utilisé par le serveur Web pour les connexions HTTP sécurisées via SSL (protocole HTTPS). Par défaut, le numéro du port HTTPS est 443 (valeur standard).

Vous pouvez souhaiter modifier ce numéro pour deux raisons principales :

- par souci de sécurité — en effet, les attaques des pirates contre les serveurs Web se concentrent généralement sur les ports TCP standard, c'est-à-dire 80 et 443.
- sous MacOS X, pour permettre aux utilisateurs "standard" de lancer le serveur Web en mode sécurisé — en effet, sous MacOS X, l'utilisation des ports TCP/IP réservés à la publication Web (0 à 1023) requiert des privilèges d'accès spécifiques : seul l'utilisateur "root" peut lancer une application utilisant ces ports. Pour que les utilisateurs autres que

“root” puissent lancer le serveur Web, une solution consiste à modifier le numéro du port TCP/IP du serveur Web.

Il est désormais possible de modifier le numéro de port HTTPS dans la page **Configuration** du thème “Web” des Préférences :



Vous pouvez passer toute valeur valide. Pour connaître les attributions standard des numéros de port TCP, reportez-vous à la section *Annexe B, Numéros des ports TCP* dans la documentation de 4D Internet Commands.

## Prise en charge des URLs inconnus

Dans les versions précédentes de 4<sup>e</sup> Dimension, la prise en charge des URLs inconnus (URLs ne correspondant à aucune page ni URLs spécial) entraînait un cas particulier : la valeur retournée dans \$1 dans les méthodes bases Sur authentification Web et Sur connexion Web ne débutait PAS par le caractère “/”.

Par exemple, si l’URL était “http://123.4.567.89/Clients/Ajouter” et qu’il n’existait pas de page “Ajouter” dans le dossier “Clients”, les méthodes base étaient appelées avec la valeur “Clients/Ajouter” dans \$1.

Ce fonctionnement atypique pouvant s’avérer problématique, il a été supprimé dans 4<sup>e</sup> Dimension 2004 : désormais, tous les URLs reçus par le serveur Web 4D débutent dans \$1 avec le caractère “/”.

**Note de compatibilité** : Certains développeurs ont pu s’appuyer sur cette particularité pour mettre en oeuvre des traitements spécifiques. Il est donc possible de conserver le fonctionnement précédent à l’aide d’une nouvelle option située dans la page **Compatibilité** des Préférences de 4<sup>e</sup> Dimension. Pour plus d’informations, reportez-vous au [paragraphe “Nouvelles options de compatibilité”, page 58](#).

## Lancement du serveur Web sous MacOS X

Sous MacOS X, l'utilisation des ports TCP/IP réservés à la publication Web (ports 0 à 1023) requiert des privilèges d'accès spécifiques : seul l'utilisateur "root" de la machine peut lancer une application utilisant ces ports. Toutefois, l'utilisation de sessions "root" présente plusieurs inconvénients, notamment du point de vue de la sécurité ainsi que pour l'accès aux documents générés au cours de la session.

D'autres solutions étaient proposées dans les versions précédentes de 4D (modification du numéro de port TCP de publication, transfert du port de réception des requêtes), mais chacune pouvait également s'avérer insatisfaisante dans certains cas.

4<sup>e</sup> Dimension 2004 propose une nouvelle possibilité : **l'ouverture d'une session root temporaire**. Le principe est le suivant : le lancement initial du serveur Web de 4<sup>e</sup> Dimension s'effectue dans une session "root" ouverte dans ce but et refermée automatiquement au bout de quelques instants. Cette nouveauté peut être utilisée avec 4<sup>e</sup> Dimension, 4D Server, 4D Client et les applications exécutables 4D Runtime Volume License (cf. [Nouveaux noms pour les applications de déploiement](#)). Notez que ce principe ne fonctionne qu'avec les bases publiées au démarrage.

Voici le détail de la séquence :

- 1 L'application 4<sup>e</sup> Dimension est exécutée au sein d'une session utilisateur classique.**
- 2 Au moment du lancement du serveur Web sur le port 80, une boîte de dialogue d'alerte informe l'utilisateur que l'opération est impossible.**  
La boîte de dialogue propose de modifier immédiatement les privilèges afin de pouvoir lancer le serveur Web. L'utilisateur devra saisir un nom et un mot de passe d'administrateur de la machine.
- 3 L'utilisateur saisit un nom et un mot de passe d'administrateur de la machine.**  
Grâce à ces informations, 4<sup>e</sup> Dimension pourra modifier les privilèges de l'application et la déclarer sous l'utilisateur "root".
- 4 L'utilisateur est invité à quitter puis relancer l'application.**  
Attention, il est nécessaire que le serveur Web soit lancé au démarrage de l'application : l'option **Publier la base au démarrage** doit être cochée dans les Préférences de l'application.

- 5 Au lancement, l'application 4<sup>e</sup> Dimension démarre dans une session root et le serveur Web est automatiquement lancé sur le port 80.**

Cette fois-ci, la publication de la base s'effectue correctement.

---

*Note* Si l'option "Autoriser SSL pour le serveur Web" est cochée, le port SSL (443 par défaut) est également ouvert.

---

- 6 Au bout de quelques instants, la session root est automatiquement refermée et remplacée par une session de l'utilisateur courant.**

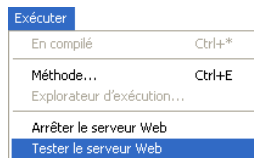
Cette opération est effectuée de manière transparente. Le serveur Web reste publié et la session utilisateur se déroule normalement.

**Important :** L'inconvénient de ce mécanisme est qu'une fois en place, vous ne pouvez plus stopper puis démarrer et redémarrer le serveur Web à volonté au cours de la session. En effet, il n'est pas possible de passer d'une session d'utilisateur standard à une session "root" (privilèges supérieurs) sans redémarrer l'application. Ce mécanisme ne fonctionne donc qu'au démarrage de l'application.

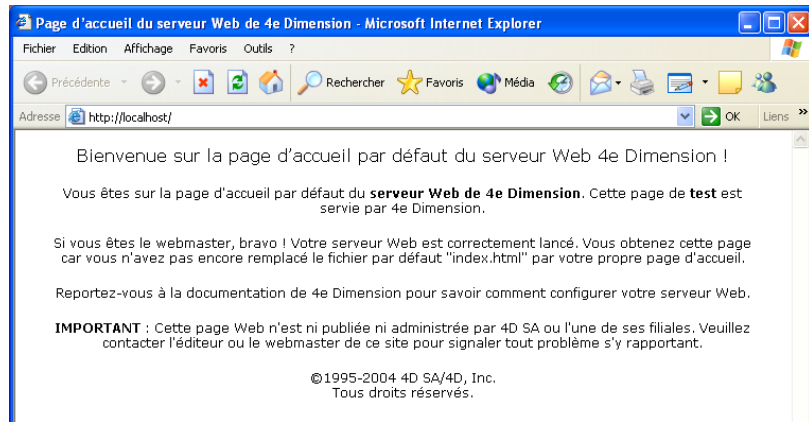
## Tester le serveur Web

Une nouvelle commande de menu de 4<sup>e</sup> Dimension 2004 permet de contrôler le fonctionnement du serveur Web intégré.

Cette commande est accessible dans le nouveau menu **Exécuter** de 4<sup>e</sup> Dimension 2004 lorsque le serveur Web est lancé :



Lorsque vous sélectionnez cette commande, la page d'accueil du site Web publié par l'application 4D s'affiche dans une fenêtre de votre navigateur par défaut :



Cette commande permet de vérifier le fonctionnement du serveur Web, l'affichage de la page d'accueil, etc.

La page est appelée via l'URL Localhost, qui est le raccourci standard désignant l'adresse IP de la machine sur laquelle est exécuté le navigateur. La commande tient compte du numéro de port TCP de publication spécifié dans les Préférences de l'application.





# 7

## Langage

Le langage de 4<sup>e</sup> Dimension version 2004 a été enrichi de nombreuses nouvelles commandes, constantes et méthodes bases. Par ailleurs, les capacités de diverses commandes existantes ont été étendues.

- Sept nouveaux thèmes accueillent les commandes correspondant aux nouvelles familles de fonctions de 4<sup>e</sup> Dimension :
  - le thème “**Formules**” contient de nouvelles commandes permettant de gérer l’éditeur de formules par programmation.
  - le thème “**Formulaire utilisateurs**” contient les commandes permettant de gérer les formulaires modifiables par les utilisateurs.
  - le thème “**Formulaires**” regroupe de nouvelles commandes permettant de gérer les formulaires créés en mode Structure ainsi que les anciennes commandes du thème “Pages formulaires” (ce thème a été supprimé de 4<sup>e</sup> Dimension 2004),
  - les commandes dédiées à la gestion du nouveau type d’objet “List box” sont intégrées dans le thème “List box”.
  - le thème “**Sauvegarde**” contient les commandes associées au nouveau module de sauvegarde de 4<sup>e</sup> Dimension,
  - le thème “**Outils**” contient diverses nouvelles commandes utilitaires ayant trait notamment à l’encodage des BLOBs et à la génération d’applications,
  - le thème “**Source de données externes**” regroupe les commandes ODBC de haut niveau permettant de connecter les applications 4<sup>e</sup> Dimension à d’autres applications ODBC.
- Les commandes du thème “**Listes hiérarchiques**” ont été modifiées pour une meilleure prise en charge des sélections d’éléments.
- La gestion des droits d’accès a été modifiée dans 4<sup>e</sup> Dimension 2004. Par conséquent, le thème “**Utilisateurs et groupes**” contient de nouvelles commandes et des commandes existantes ont été modifiées.

- Le thème “Fenêtres” a été enrichi afin de fournir aux développeurs les moyens de définir des fenêtres de dimensions variables.
- Le thème “Liens” s’est enrichi de nouvelles commandes permettant une gestion plus précise du statut des liens.
- De nouveaux événements formulaires sont désormais disponibles dans le thème “Événements formulaires” pour la gestion des survols (*roll over*), des boutons 3D (bevels), des objets List box, des listes d’enregistrements et des listes hiérarchiques.
- Le thème “Interface utilisateur” contient les nouvelles commandes LIRE TITRES TABLES, LIRE TITRES CHAMPS et DEFILER LIGNES ; la commande Pop up menu a été modifiée.
- La nouvelle commande EDITER ELEMENT a été ajoutée dans le thème “Gestion de la saisie”.
- Les commandes du thème “Impressions” peuvent désormais être utilisées sur le poste serveur.
- La nouvelle commande AFFICHER NOTIFICATION a été ajoutée dans le thème “Messages”.
- Le fonctionnement par défaut de la commande DIALOGUE (thème “Saisie”) a été modifié.
- Le thème “Documents système” contient la nouvelle commande Selectionner document.
- Le thème “Propriétés des objets” contient une nouvelle commande et plusieurs commandes ont été modifiées.
- Le thème “Environnement 4D” contient la nouvelle commande OUVRIER PREFERENCE 4D, permettant d’ouvrir n’importe quelle page de la boîte de dialogue des Préférences.
- Le thème “Environnement système” contient deux nouvelles commandes permettant à 4<sup>e</sup> Dimension d’interagir avec son environnement : FIXER VARIABLE ENVIRONNEMENT et LANCER PROCESS EXTERNE.
- Le thème “Sélections” contient les nouvelles commandes LIRE ENREGISTREMENTS MARQUES et Numero de ligne affichee ; les commandes MARQUER ENREGISTREMENTS, MODIFIER SELECTION, VISUALISER SELECTION ainsi que ALLER DANS SELECTION ont été modifiées.

- Le thème “Serveur Web” comporte désormais trois nouvelles commandes : **TRAITER BALISES HTML**, **LIRE CORPS HTTP** et **ENVOYER DONNEES HTTP** ; la commande **LIRE VARIABLES FORMULAIRE WEB** a été modifiée.
- Les nouvelles commandes **Compter dans tableau** et **AJOUTER A TABLEAU** ont été ajoutées dans le thème “Tableaux”.
- La nouvelle commande **LIRE CORRESPONDANCE PORT SERIE** a été ajoutée dans le thème “Communications”.
- Le thème “XML” a été profondément remanié afin d’accueillir les nouvelles commandes du standard SAX ainsi que les commandes DOM permettant de créer des arbres XML.
- La nouvelle commande **FIXER OPTION WEB SERVICE** a été ajoutée dans le thème “Web Services (Client)”.

## Formules

L’interface de l’Editeur de formules a été modifiée dans 4<sup>e</sup> Dimension 2004 afin de faciliter la construction de formules personnalisées tout en permettant un meilleur contrôle par le développeur. Seules les commandes utilisables dans le cadre d’une formule sont affichées. Ces modifications sont détaillées dans le [paragraphe “Editeur de formules”](#), page 35.

Parallèlement, de nouvelles commandes ont été créées afin de fournir au développeur le moyen d’afficher directement l’éditeur de formules (**EDITER FORMULE**) et de filtrer les méthodes adaptées à l’écriture de formules (**FIXER METHODES AUTORISEES** et **LIRE METHODES AUTORISEES**).

### **FIXER METHODES AUTORISEES**

**FIXER METHODES AUTORISEES**(tabMéthodes)

Paramètres	Type	Description
tabMéthodes	Tableau alpha	→ Tableau de noms de méthodes

La commande **FIXER METHODES AUTORISEES** permet de définir les méthodes qui seront affichées dans l’éditeur de formules pour le process courant. Les méthodes désignées apparaîtront à la fin de la liste des commandes et pourront être utilisées dans les formules. Par défaut (si vous n’utilisez pas cette commande), aucune méthode n’est

utilisable dans l'éditeur de formules. Si une formule utilise un nom de méthode non autorisée, une erreur de syntaxe est générée et la formule ne peut pas être validée.

Passez dans le paramètre *tabMéthodes* le nom du tableau contenant la liste des méthodes à proposer dans l'éditeur de formules. Le tableau doit avoir été préalablement défini.

Vous pouvez utiliser le caractère "joker" (@) dans les noms des méthodes afin de définir un ou plusieurs groupe(s) de méthodes autorisées.

---

*Note* Si vous souhaitez que l'utilisateur puisse appeler des commandes 4D non autorisées par défaut ou des commandes de plug-ins, vous devez utiliser des méthodes spécifiques chargées d'exécuter ces commandes.

---

- ▼ Autorisation de toutes les méthodes dont le nom débute par "formule" et de la méthode "Total\_général" dans l'éditeur de formules :

```
TABLEAU ALPHA(15;tabméthodes;2)
tabméthodes{1}:="formule@"
tabméthodes{2}:="Total_général"
FIXER METHODES AUTORISEES(tabméthodes)
```

*Référence* : [LIRE METHODES AUTORISEES](#)

## LIRE METHODES AUTORISEES

LIRE METHODES AUTORISEES(tabMéthodes)

Paramètres	Type	Description
tabMéthodes	Tableau Alpha ←	Tableau de noms de méthodes

La commande LIRE METHODES AUTORISEES retourne dans le tableau *tabMéthodes* le nom des méthodes "autorisées" dans l'éditeur de formules, c'est-à-dire pouvant être utilisées lors de l'écriture d'une formule — ces méthodes sont listées à la fin de la liste des commandes dans l'éditeur.

Par défaut, aucune méthode n'est utilisable dans l'éditeur de formules. Les méthodes doivent avoir été explicitement autorisées via la commande [FIXER METHODES AUTORISEES](#). Si cette commande n'a pas été exécutée, LIRE METHODES AUTORISEES retourne un tableau vide.

LIRE METHODES AUTORISEES retourne précisément ce qui a été passé à la commande [FIXER METHODES AUTORISEES](#), c'est-à-dire un tableau alpha (la commande crée et dimensionne le tableau). En outre, si le caractère "joker" (@) a été utilisé pour désigner un groupe de

méthodes, la chaîne contenant le caractère @ est retournée (et non les noms des méthodes du groupe).

Cette commande est utile pour préserver le paramétrage de l'ensemble courant de méthodes autorisées avant l'exécution d'une formule dans un contexte spécifique (par exemple un état rapide).

- ▼ Cet exemple permet d'autoriser un ensemble de méthodes spécifiques pour la création d'un état rapide :

```
`Stockage du paramétrage courant
LIRE METHODES AUTORISEES(tabméthodes)
```

```
`Définition des méthodes pour l'état
méthodes_Etats:="Etats_@"
FIXER METHODES AUTORISEES(méthodes_Etats)
QR ETAT([Personnes];"MonEtat")
```

```
`Rétablissement des paramètres courants
FIXER METHODES AUTORISEES(tabméthodes)
```

**Référence :** [FIXER METHODES AUTORISEES](#)

## EDITER FORMULE

EDITER FORMULE(table;formule)

Paramètres	Type	Description
table	Table	→ Table à afficher par défaut dans l'éditeur de formules
formule	Variable chaîne	→ Variable contenant la formule à afficher dans l'éditeur de formules ou "" pour uniquement afficher l'éditeur ← Formule validée par l'utilisateur

La commande EDITER FORMULE affiche l'éditeur de formules afin de permettre à l'utilisateur d'écrire ou de modifier une formule. L'éditeur contient à l'ouverture :

- dans la liste de gauche, les champs de la table désignée par le paramètre *table*,
- dans la zone de formule, la formule contenue dans la variable *formule*. Si vous avez passé une chaîne vide dans *formule*, l'éditeur de formules est affiché sans formule.

L'utilisateur peut modifier la *formule* affichée et la sauvegarder. Il peut également en écrire ou en charger une nouvelle. Dans tous les cas,

lorsque l'utilisateur valide la boîte de dialogue, la variable système *OK* prend la valeur 1 et la variable *formule* contient la formule définie. Si l'utilisateur annule la boîte de dialogue, la variable système *OK* prend la valeur 0 et *formule* est inchangée.

---

*Note* Si la *formule* fait appel à des méthodes qui n'ont pas été préalablement "autorisées" dans l'éditeur de formules à l'aide de la commande **FIXER METHODES AUTORISEES**, une erreur de syntaxe est générée et il n'est pas possible de valider la boîte de dialogue.

---

A noter qu'au moment de la validation de la boîte de dialogue, la commande n'exécute pas la formule définie, seul le contenu de la variable *formule* est validé et mis à jour. Si vous voulez exécuter la *formule*, vous devez utiliser la commande EXECUTER.

- ▼ Affichage de l'éditeur avec la table [Salaires] et sans formule pré-saisie, puis exécution de la formule sur la sélection courante :

```
$maFormule:=""  
EDITER FORMULE([Salaires];$maFormule)  
Si (OK=1)  
  APPLIQUER A SELECTION([Salaires];EXECUTER($maFormule))  
Fin de si
```

*Référence* : **FIXER METHODES AUTORISEES**

## Formulaire utilisateurs

Dans 4<sup>e</sup> Dimension 2004, le développeur peut proposer aux utilisateurs de créer ou de modifier des formulaires personnalisés. Les "formulaire utilisateurs" sont ensuite utilisables comme n'importe quel formulaire de 4<sup>e</sup> Dimension.

### Principes de fonctionnement

Les formulaires utilisateurs sont basés sur des formulaires 4<sup>e</sup> Dimension standard créés par le développeur en mode Structure (appelé formulaire "source" ou formulaire "développeur"), auxquels la propriété **Modifiable par l'utilisateur** a été appliquée. Un éditeur de formulaires simplifié (appelé via la nouvelle commande **MODIFIER FORMULAIRE**) permet aux utilisateurs de modifier l'apparence du formulaire, d'ajouter des objets graphiques, de masquer des éléments, etc. — le développeur peut contrôler les actions autorisées. Pour plus d'informations sur l'éditeur de formulaires utilisateurs,

reportez-vous au [paragraphe “Editeur de formulaires utilisateurs”](#), page 29.

Les formulaires utilisateurs peuvent être employés de deux manières différentes :

- L'utilisateur modifie le formulaire “source” pour l'adapter à ses besoins à l'aide de la commande **MODIFIER FORMULAIRE**.  
Le formulaire utilisateur est conservé en local et est utilisé automatiquement à la place du formulaire original.  
Ce fonctionnement répond aux besoins pour le développeur de paramétrer sur site des boîtes de dialogue, par exemple pour coller le logo de l'entreprise dans les formulaires, masquer des champs inutiles, etc.
- Le formulaire “source” sert de modèle de base que l'utilisateur peut dupliquer à loisir pour générer autant de copies que nécessaire (via la nouvelle commande **CREER FORMULAIRE UTILISATEUR**). Chaque copie est librement paramétrable (contenu, nom, etc.) à l'aide de la commande **MODIFIER FORMULAIRE**. Le nom de chaque formulaire utilisateur doit simplement être unique. Pour utiliser ces copies, il est nécessaire d'utiliser les commandes **FORMULAIRE ENTREE** et **FORMULAIRE SORTIE** qui acceptent désormais le nom d'un formulaire utilisateur comme paramètre.  
Ce fonctionnement répond par exemple aux besoins de génération d'états personnalisés pour l'utilisateur.

## Formulaires utilisateurs et bibliothèques d'objets

Il est possible d'insérer des objets provenant d'une bibliothèque d'objets 4<sup>e</sup> Dimension dans l'éditeur de formulaires utilisateurs. Pour plus d'informations sur les bibliothèques d'objets, reportez-vous au [paragraphe “Bibliothèques d'objets”](#), page 175.

Lorsqu'elles sont utilisées dans le contexte de l'éditeur de formulaires utilisateurs, les bibliothèques d'objets permettent de coller des objets avec leurs propriétés graphiques et leurs actions automatiques. Les objets auxquels une méthode est associée n'apparaissent pas dans la bibliothèque.

Attention, il est du ressort du développeur de vérifier que l'ajout des objets d'une bibliothèque n'est pas incompatible avec le formulaire utilisateur (et ses objets) au niveau des noms, des variables et des types.

## Stockage et gestion des formulaires utilisateurs

Les mécanismes des formulaires utilisateurs fonctionnent avec les bases compilées et interprétées, avec 4<sup>e</sup> Dimension, 4D Server ou 4D Runtime. En mode client/serveur, les formulaires modifiés par l'utilisateur sont disponibles sur tous les postes.

4D assure automatiquement la gestion des modifications des formulaires. Lorsqu'un formulaire est déclaré **Modifiable par l'utilisateur**, il est verrouillé en mode Structure. Le développeur doit explicitement cliquer sur l'icône de déverrouillage afin de pouvoir accéder aux objets du formulaire. Cette opération rend automatiquement obsolètes les formulaires utilisateurs liés, qui devront alors être régénérés.

Lorsqu'un formulaire "source" est supprimé, les formulaires utilisateurs liés sont supprimés.

Les formulaires utilisateurs sont stockés dans un fichier indépendant suffixé .4DA, placé à côté du fichier de structure principal (.4DB / .4DC). Le fonctionnement de ce fichier est transparent : 4<sup>e</sup> Dimension utilise un formulaire utilisateur lorsqu'il existe (la nouvelle commande [LISTE FORMULAIRES UTILISATEURS](#) permet de connaître à tout moment les formulaires utilisateurs valides). Lorsqu'un formulaire utilisateur est obsolète, il est supprimé et 4<sup>e</sup> Dimension utilise par défaut le formulaire source.

En client/serveur, le fichier .4DA est distribué sur les postes clients suivant les mêmes règles que le fichier de structure principal.

Ce principe permet en outre de conserver les formulaires utilisateurs non obsolètes en cas de mise à jour de la structure par le développeur.

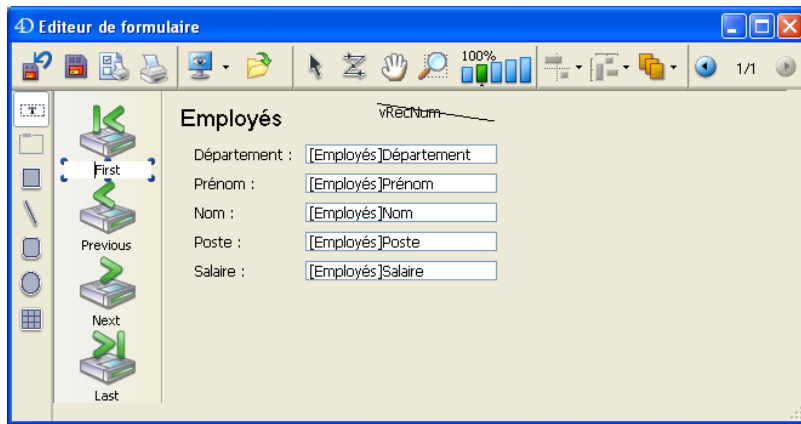
## MODIFIER FORMULAIRE

MODIFIER FORMULAIRE (table; formulaire{; formUtilisateur}{; bibliothèque})

Paramètres	Type	Description
table	Table	→ Table du formulaire à modifier
formulaire	Chaîne	→ Nom du formulaire à modifier
formUtilisateur	Chaîne	→ Nom du formulaire utilisateur à modifier
bibliothèque	Chaîne	→ Nom long de la bibliothèque d'objets utilisable



La commande MODIFIER FORMULAIRE ouvre le formulaire désigné par les paramètres *table*, *formulaire* ainsi que, facultativement, *formUtilisateur* dans l'éditeur de formulaires utilisateurs :



*Note* La fenêtre de l'éditeur ne s'ouvre que si elle est la première fenêtre du process. Autrement dit, il sera généralement nécessaire d'ouvrir un nouveau process pour afficher l'éditeur.

Si l'utilisateur sauvegarde les modifications éventuellement effectuées dans cet éditeur, la variable *OK* prend la valeur 1.

Si vous ne passez pas le paramètre *formUtilisateur* et s'il n'existe pas déjà un formulaire utilisateur lié à *formulaire*, le formulaire source est affiché dans l'éditeur. Le formulaire modifié est ensuite dupliqué dans le fichier de structure utilisateur (.4DA) et sera utilisé en remplacement de *formulaire*.

Si un formulaire utilisateur avait déjà été généré à partir de *formulaire* à l'aide de cette commande, le formulaire utilisateur s'affiche dans l'éditeur. Si vous souhaitez dans ce cas repartir du formulaire source, vous devez au préalable supprimer le formulaire utilisateur à l'aide de la commande **SUPPRIMER FORMULAIRE UTILISATEUR**.

Le paramètre *formUtilisateur* permet de désigner un formulaire utilisateur (créé à l'aide de la commande **CREER FORMULAIRE UTILISATEUR**) à modifier. Dans ce cas, ce formulaire est affiché dans l'éditeur.

Passez dans le paramètre facultatif *bibliothèque* le chemin d'accès complet de la bibliothèque d'objets que l'utilisateur sera autorisé à utiliser pour personnaliser le formulaire. En mode client/serveur, la

bibliothèque doit se trouver dans le dossier **Extras** de la base de données, au même niveau que le dossier Plug-ins, afin qu'elle soit disponible sur tous les postes clients. Si la *bibliothèque* est valide, elle est ouverte avec la fenêtre du formulaire. Pour plus d'informations sur les bibliothèques d'objets, reportez-vous au [paragraphe "Bibliothèques d'objets"](#), page 175.

La variable système *OK* retourne 1 si l'opération s'est déroulée correctement. Elle prend la valeur 0 et une erreur est générée si :

- le formulaire n'a pas été déclaré modifiable par l'utilisateur en mode Structure ou n'existe pas,
- le formulaire est déjà ouvert en modification dans un autre process,
- l'utilisateur ne possède pas les droits d'accès adéquats.

Vous pouvez intercepter ces erreurs à l'aide d'une méthode installée par la commande APPELER SUR ERREUR.

- ▼ Voici une méthode projet permettant à l'utilisateur de modifier un formulaire de dialogue :

**MODIFIER FORMULAIRE**([Dialogs];"Welcome";"Bib\_Logos.4il")

## CREER FORMULAIRE UTILISATEUR

CREER FORMULAIRE UTILISATEUR(table; formulaire; formUtilisateur)

Paramètres	Type	Description
table	Table	→ Table du formulaire source
formulaire	Chaîne	→ Nom du formulaire source
formUtilisateur	Chaîne	→ Nom du nouveau formulaire utilisateur

La commande CREER FORMULAIRE UTILISATEUR duplique le *formulaire* 4<sup>e</sup> Dimension dont la table et le nom sont passés en paramètres et crée un nouveau formulaire utilisateur nommé *formUtilisateur*.

Une fois créé, le formulaire *formUtilisateur* pourra être modifié à l'aide de la commande **MODIFIER FORMULAIRE**. Cette commande permet de créer N formulaires utilisateurs (par exemple divers formulaires d'états) à partir d'un même formulaire source.

La variable *OK* retourne 1 si l'opération s'est déroulée correctement. Elle prend la valeur 0 et une erreur est générée si :

- *formulaire* est déjà un formulaire utilisateur,
- le nom du formulaire utilisateur *formUtilisateur* est identique à celui du formulaire source ou d'un formulaire utilisateur existant,

- l'utilisateur ne possède pas les droits d'accès adéquats.

Vous pouvez intercepter ces erreurs à l'aide d'une méthode installée par la commande APPELER SUR ERREUR.

## LISTE FORMULAIRES UTILISATEURS

LISTE FORMULAIRES UTILISATEURS(table; formulaire; tabFormUtilisateurs))

Paramètres	Type	Description
table	Table	→ Table du formulaire source
formulaire	Chaîne	→ Nom du formulaire source
tabFormUtilisateurs	Tab Alpha	← Noms des formulaires utilisateurs issus du formulaire source

Cette commande remplit le tableau *tabFormUtilisateurs* avec les noms des formulaires utilisateurs issus du formulaire développeur désigné par les paramètres *table* et *formulaire*.

Si le formulaire utilisateur a été créé directement via la commande MODIFIER FORMULAIRE, *tabFormUtilisateurs* contient comme seul élément une chaîne vide ("").

Si aucun formulaire utilisateur n'existe pour le formulaire développeur spécifié, le tableau est retourné vide.

## SUPPRIMER FORMULAIRE UTILISATEUR

SUPPRIMER FORMULAIRE UTILISATEUR(table; formulaire; formUtilisateur)

Paramètres	Type	Description
table	Table	→ Table du formulaire utilisateur
formulaire	Chaîne	→ Nom du formulaire source
formUtilisateur	Chaîne	→ Nom du formulaire utilisateur

Cette commande permet de supprimer le formulaire utilisateur désigné par les paramètres *table*, *formulaire* et *formUtilisateur*.

Si le formulaire utilisateur a été créé directement via la commande MODIFIER FORMULAIRE, passez une chaîne vide ("" dans *formUtilisateur*.

Si le formulaire utilisateur est correctement supprimé, la variable OK retourne 1. Elle prend la valeur 0 et une erreur est générée si :

- le formulaire utilisateur n'existe pas,
- l'utilisateur ne possède pas les droits d'accès adéquats.

Vous pouvez intercepter ces erreurs à l'aide d'une méthode installée par la commande APPELER SUR ERREUR.

## Codes d'erreurs des formulaires utilisateurs

Voici la liste des nouveaux codes d'erreurs 4<sup>e</sup> Dimension relatifs à l'utilisation des commandes de gestion des formulaires utilisateurs :

- 9750 Le formulaire source n'est pas modifiable
- 9751 Le formulaire source n'est pas accessible pour l'utilisateur
- 9752 Le formulaire utilisateur ne peut pas être créé
- 9753 Le formulaire source n'existe pas
- 9754 Cette commande ne peut pas être appelée depuis une fenêtre de dialogue
- 9755 Le formulaire utilisateur n'a pas de nom
- 9756 Il n'y a pas de fichier de structure utilisateur
- 9757 Le formulaire utilisateur n'existe pas
- 9758 Le formulaire utilisateur existe déjà
- 9759 La bibliothèque d'objets n'a pas pu être ouverte

## Listes hiérarchiques

La gestion programmée des listes hiérarchiques a été modifiée dans 4<sup>e</sup> Dimension 2004. En particulier, la prise en charge des sélections d'éléments multiples a été améliorée et l'alternative de sélection position/référence a été généralisée.

Ces améliorations se traduisent par la modification des commandes **SELECTIONNER ELEMENTS PAR POSITION (SELECTIONNER ELEMENT)**, **SELECTIONNER ELEMENTS PAR REFERENCE**, **CHANGER PROPRIETES LISTE**, **LIRE PROPRIETES LISTE**, **CHANGER PROPRIETES ELEMENT**, **LIRE PROPRIETES ELEMENT**, **CHANGER ELEMENT**, **INFORMATION ELEMENT**, **Element parent**, **Elements selectionnes (Element selectionne)** et **Nombre elements**.

A noter également l'ajout de la commande **EDITER ELEMENT** dans le thème "**Gestion de la saisie**". Cette nouvelle commande permet par programmation de passer en "mode édition" un élément de liste, notamment de liste hiérarchique.

Enfin, de nouveaux événements formulaires permettent une meilleure prise en charge des actions utilisateurs (cf. [paragraphe "Événements formulaire liés aux listes hiérarchiques"](#), page 307).

## SELECTIONNER ELEMENTS PAR POSITION (SELECTIONNER ELEMENT)

SELECTIONNER ELEMENTS PAR POSITION(liste; positionElém{; *tabPositions*})

Paramètres	Type	Description
liste	RéfListe	→ Numéro de référence de liste
positionElém	Numérique	→ Position de l'élément dans la ou les liste(s) déployée(s)
<i>tabPositions</i>	Tableau numérique	→ Tableau de positions dans la ou les liste(s) déployée(s)

**Note préliminaire** : SELECTIONNER ELEMENTS PAR POSITION est le nouveau nom de la commande SELECTIONNER ELEMENT, déjà présente dans les versions précédentes de 4<sup>e</sup> Dimension. Cette commande a été renommée par souci de clarification en rapport avec la commande SELECTIONNER ELEMENTS PAR REFERENCE et pour mettre en évidence la possibilité de multi-sélections dans la version 2004 de 4<sup>e</sup> Dimension.

La commande SELECTIONNER ELEMENTS PAR POSITION admet désormais un paramètre supplémentaire (facultatif), *tabPositions*. Ce paramètre permet de sélectionner simultanément plusieurs éléments au sein de la *liste*.

Vous devez passer dans *tabPositions* un tableau dont chaque ligne indique la position d'un élément à sélectionner.

**Important** : Pour que plusieurs éléments puissent être sélectionnés simultanément dans une liste hiérarchique (manuellement ou par programmation), la nouvelle propriété *multiSélections* doit avoir été activée pour cette liste. Cette propriété est définie via la commande **CHANGER PROPRIETES LISTE**.

L'élément désigné par le paramètre *positionElém* désigne parmi la sélection résultante le nouvel élément courant de la liste. Il peut appartenir ou non à l'ensemble d'éléments définis par le tableau. L'élément courant est notamment celui qui passe en mode édition si la commande **EDITER ELEMENT** est utilisée.

- ▼ Cet exemple provoque la sélection des 2e, 3e et 5e éléments de la liste hiérarchique :

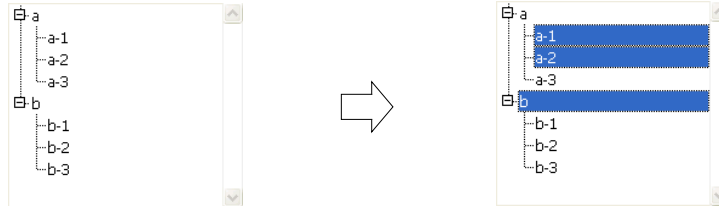
**CHANGER PROPRIETES LISTE**(hList;0;0;18;0;1)

  `Il est impératif de passer 1 en dernier paramètre pour autoriser  
  `les multi-sélections

**TABLEAU ENTIER LONG**(\$tab;3)

\$tab{1}:=2

```
$tab{2}:=3
$tab{3}:=5
SELECTIONNER ELEMENTS PAR POSITION(hList;3;$tab)
`Le 3e élément est désigné comme élément courant
REDESSINER LISTE(hList)
```



## SELECTIONNER ELEMENTS PAR REFERENCE

SELECTIONNER ELEMENTS PAR REFERENCE(liste; refElém{; tabRéfs})

Paramètres	Type	Description
liste	RéfListe	→ Numéro de référence de liste
refElém	Entier long	→ Numéro de référence d'élément
tabRéfs	Tableau entier long	→ Tableau de numéros de référence d'éléments

**Note préliminaire** : SELECTIONNER ELEMENTS PAR REFERENCE est le nouveau nom de la commande SELECTIONNER ELEMENT PAR REFERENCE, déjà présente dans les versions précédentes de 4<sup>e</sup> Dimension. Cette commande a été renommée pour mettre en évidence la possibilité de multi-sélections dans la version 2004 de 4<sup>e</sup> Dimension.

La commande SELECTIONNER ELEMENTS PAR REFERENCE admet désormais un paramètre supplémentaire (facultatif), *tabRéfs*. Ce paramètre permet de sélectionner simultanément plusieurs éléments au sein de la *liste*.

Vous devez passer dans *tabRéfs* un tableau dont chaque ligne indique la référence absolue d'un élément à sélectionner.

**Important** : Pour que plusieurs éléments puissent être sélectionnés simultanément dans une liste hiérarchique (manuellement ou par programmation), la nouvelle propriété *multiSélections* doit avoir été activée pour cette liste. Cette propriété est définie via la commande **CHANGER PROPRIETES LISTE** (cf. ci-dessous).

L'élément désigné par le paramètre *refElém* désigne parmi la sélection résultante le nouvel élément courant de la liste. Il peut appartenir ou

non à l'ensemble d'éléments définis par le tableau. L'élément courant est notamment celui qui passe en mode édition si la commande **EDITER ELEMENT** est utilisée.

## CHANGER PROPRIETES LISTE

CHANGER PROPRIETES LISTE(liste; apparence{; icône{; hauteurLigne{; doubleClic{; multiSélections{; modifiable{}}})

Paramètres	Type	Description
liste	RéfListe	→ Numéro de référence de liste
apparence	Numérique	→ Style graphique de la liste
icône	Numérique	→ Référence de ressource MacOS 'cicn' ou 0 = icône par défaut
hauteurLigne	Numérique	→ Hauteur minimale de ligne
doubleClic	Entier long	→ Déploiement/contraction sur double-clic : 0=autoriser (défaut), 1=interdire
<i>multiSélections</i>	<i>Entier long</i>	→ <i>Sélections multiples : 0=interdire (défaut), 1=autoriser</i>
<i>modifiable</i>	<i>Entier long</i>	→ <i>0=Énumération non modifiable 1=Énumération modifiable (défaut)</i>

La commande CHANGER PROPRIETES LISTE admet deux nouveaux paramètres facultatifs, *multiSélections* et *modifiable*.

- ***multiSélections*** : ce paramètre permet d'indiquer si la *liste* doit accepter les sélections multiples.

Par défaut, comme dans les versions précédentes de 4<sup>e</sup> Dimension, il n'est pas possible de sélectionner simultanément plusieurs éléments d'une liste hiérarchique. Si vous souhaitez que cette fonction soit disponible pour la liste, passez la valeur 1 dans le paramètre *multiSélections*. Dans ce cas, les sélections multiples peuvent être effectuées :

- manuellement, à l'aide des combinaisons de touches **Maj+clik** pour une sélection continue ou **Ctrl+clik** (Windows) / **Commande+clik** (MacOS) pour une sélection discontinue,
- par programmation, à l'aide des commandes SELECTIONNER ELEMENTS PAR POSITION et SELECTIONNER ELEMENTS PAR REFERENCE.

Si vous passez 0 ou omettez le paramètre *multiSélections*, le fonctionnement par défaut est appliqué.

- ***modifiable*** : ce paramètre permet d'indiquer si la liste sera modifiable par l'utilisateur lorsqu'elle sera affichée sous forme d'énumération associée à un champ ou une variable en saisie. Lorsque l'énumération

est modifiable, un bouton **Modifier** est inséré dans la fenêtre d'énumération et l'utilisateur peut ajouter, supprimer et trier les valeurs via un éditeur spécifique.

Si vous passez 0 ou omettez le paramètre *modifiable*, l'énumération sera modifiable par l'utilisateur ; si vous passez 1, elle ne sera pas modifiable.

**Référence** : [LIRE PROPRIETES LISTE](#).

## LIRE PROPRIETES LISTE

LIRE PROPRIETES LISTE(liste; apparence{; icône{; hauteurLigne{; doubleClic{; multiSélections{; modifiable{}}})

Paramètres	Type	Description
liste	RéfListe	→ Numéro de référence de liste
apparence	Numérique	← Style graphique de la liste
icône	Numérique	← Référence de ressource MacOS 'cicn'
hauteurLigne	Numérique	← Hauteur minimale de ligne
doubleClic	Entier long	← Déploiement/contraction sur double-clic : 0=autorisé, 1=interdit
<i>multiSélections</i>	<i>Entier long</i>	← <i>Sélections multiples : 0=interdit, 1=autorisé</i>
<i>modifiable</i>	<i>Entier long</i>	← <i>0=Enumération non modifiable, 1=Enumération modifiable</i>

La commande LIRE PROPRIETES LISTE retourne deux paramètres facultatifs supplémentaires, *multiSélections* et *modifiable*.

- Si le paramètre *multiSélections* vaut 0, la sélection multiple d'éléments (manuelle ou par programmation) n'est pas possible dans la *liste*. S'il vaut 1, la sélection multiple est permise.
- Si le paramètre *modifiable* vaut 0, la liste est modifiable lorsqu'elle est affichée sous forme d'énumération en mode Utilisation. S'il vaut 1, la liste n'est pas modifiable.

Pour plus d'informations sur ces paramètres, reportez-vous à la description de la [commande CHANGER PROPRIETES LISTE](#), page 263.

**Référence** : [CHANGER PROPRIETES LISTE](#).



## CHANGER PROPRIETES ELEMENT

CHANGER PROPRIETES ELEMENT(liste; réfÉlément | \*; saisissable; styles; icône(; couleur))

Paramètres	Type	Description
liste	RéfListe	→ Numéro de référence de liste
réfÉlément   *	Entier long   *	→ Numéro de référence d'élément ou 0 pour le dernier élément ajouté ou * pour l'élément courant de la liste
saisissable	Booléen	→ Vrai=Saisissable, Faux=Non saisissable
styles	Numérique	→ Style de police pour l'élément
icône	Numérique	→ Numéro de ressource MacOS 'cicn' ou 65536 + numéro de ressource MacOS PICT ou 131072 + numéro de référence d'image
couleur	Entier long	→ Valeur de couleur RVB ou -1 = rétablir couleur originale

La commande CHANGER PROPRIETES ELEMENT a subi deux modifications : d'une part, elle accepte désormais la valeur \* comme référence d'élément et d'autre part, elle admet un nouveau paramètre permettant de modifier la couleur de l'élément désigné.

- Vous pouvez désormais passer \* en deuxième paramètre : dans ce cas, la commande s'appliquera à l'élément courant de la liste. Si plusieurs éléments sont sélectionnés manuellement, l'élément courant est celui qui a été sélectionné en dernier. Si aucun élément n'est sélectionné, la commande n'a pas d'effet.
- La commande admet le paramètre (facultatif) *couleur*, permettant de modifier la couleur du texte de l'élément désigné. La couleur doit être définie sous forme de couleur RVB, c'est-à-dire un entier long de 4 octets au format 0x00RRVVBB. Pour plus d'informations sur ce format, reportez-vous à la description de la commande FIXER COULEURS RVB dans le manuel *Langage* de 4<sup>e</sup> Dimension. Passez -1 dans le paramètre *couleur* pour rétablir la couleur d'origine de l'élément.
- ▼ L'exemple suivant passe le texte de l'élément courant de *liste* en gras et en rouge vif :

```
CHANGER PROPRIETES ELEMENT(liste,*;Vrai;Gras;0;0x00FF0000)
REDESSINER LISTE(liste)
```

*Référence* : LIRE PROPRIETES ELEMENT.

## LIRE PROPRIETES ELEMENT

LIRE PROPRIETES ELEMENT(liste; réfElément | \*; saisissable; styles; icône{; couleur})

Paramètres	Type	Description
liste	RéfListe	→ Numéro de référence de liste
réfElément   *	Entier long   *	→ Numéro de référence d'élément ou 0 pour le dernier élément ajouté ou * pour l'élément courant de la liste
saisissable	Booléen	← Vrai=Saisissable, Faux=Non saisissable
styles	Numérique	← Style de police pour l'élément
icône	Numérique	← Numéro de ressource MacOS 'cicn' ou 65536 + numéro de ressource MacOS PICT ou 131072 + numéro de référence d'image
couleur	Entier long	← Valeur de couleur RVB

La commande LIRE PROPRIETES ELEMENT a subi deux modifications : d'une part, elle accepte la valeur \* comme référence d'élément et d'autre part, elle admet un nouveau paramètre retournant la couleur de l'élément désigné.

- Vous pouvez désormais passer \* en deuxième paramètre : dans ce cas, la commande s'appliquera à l'élément courant de la liste. Si plusieurs éléments sont sélectionnés manuellement, l'élément courant est celui qui a été sélectionné en dernier. Si aucun élément n'est sélectionné, la commande ne retourne pas de valeur.
- La commande admet le paramètre (facultatif) *couleur*, retournant la couleur du texte de l'élément désigné. Pour plus d'informations sur ce paramètre, reportez-vous à la description de la [commande CHANGER PROPRIETES ELEMENT](#), page 265.

**Référence :** [CHANGER PROPRIETES ELEMENT](#).

**CHANGER ELEMENT** CHANGER ELEMENT(liste; réfElément | \*; textElément; nouvRéf{; sous\_Liste; déployée})

Paramètres	Type	Description
liste	RéfListe	→ Numéro de référence de liste
réfElément   *	Entier long   *	→ Numéro de référence d'élément ou 0 pour le dernier élément ajouté ou * pour l'élément courant de la liste
textElément	Alpha	→ Nouveau libellé de l'élément
nouvRéf	Numérique	→ Nouveau numéro de référence de l'élément
sous_Liste	RéfListe	→ Nouvelle sous-liste rattachée à l'élément ou 0=pas de sous-liste (détacher sous-liste courante) ou -1=pas de changement
déployée	Booléen	→ Sous-liste déployée/contractée

La commande CHANGER ELEMENT accepte désormais la valeur \* comme référence d'élément : dans ce cas, la commande s'applique à l'élément courant de la liste. Si plusieurs éléments sont sélectionnés manuellement, l'élément courant est celui qui a été sélectionné en dernier. Si aucun élément n'est sélectionné, la commande n'a pas d'effet.

**Référence :** INFORMATION ELEMENT

## INFORMATION ELEMENT

INFORMATION ELEMENT(liste; positionElém | \*; réfElément; textElément{; sous\_Liste{; déployée}})

Paramètres	Type	Description
liste	RéfListe	→ Numéro de référence de liste
positionElém   *	Num   *	→ Position de l'élément dans la ou les liste(s) déployée(s) ou * pour l'élément courant de la liste
réfElément	Entier long	← Numéro de référence d'élément
textElément	Alpha	← Libellé de l'élément
sous_Liste	RéfListe	← Numéro de référence de sous-liste (s'il y en a)
déployée	Booléen	← Sous-liste déployée/contractée

La commande INFORMATION ELEMENT accepte désormais la valeur \* comme position d'élément : dans ce cas, la commande s'applique à l'élément courant de la liste.

Si plusieurs éléments sont sélectionnés manuellement, l'élément courant est celui qui a été sélectionné en dernier. Si aucun élément n'est sélectionné, la commande retourne des valeurs vides.

**Référence :** CHANGER ELEMENT

## Element parent

Element parent(liste; réfElément | \*) → Numérique

Paramètres	Type	Description
liste	RéfListe	→ Numéro de référence de liste
réfElément   *	Entier long   *	→ Numéro de référence d'élément ou * pour l'élément courant de la liste
Résultat	Entier long	← Numéro de référence de l'élément parent ou 0 s'il n'y en a pas

La commande Element parent accepte désormais la valeur \* comme référence d'élément : dans ce cas, la commande s'applique à l'élément courant de la liste. Si plusieurs éléments sont sélectionnés, l'élément courant est celui qui a été sélectionné en dernier. Si aucun élément n'est sélectionné, la commande ne retourne pas de valeur.

## Elements selectionnes (Element selectionne)

Elements selectionnes(liste{; tabEléments}{; \*}) → Entier long

Paramètres	Type	Description
liste	RéfListe	→ Numéro de référence de liste
tabEléments	Tab Entier long	← Si * omis : Tableau contenant les positions des éléments sélectionnés parmi la ou les liste(s) déployée(s) Si * passé : Tableau contenant les références des éléments sélectionnés
*	*	→ Si omis : Position(s) d'élément(s) Si passé : Référence(s) d'élément(s)
Résultat	Numérique	← Si * omis : Position de l'élément sélectionné parmi la ou les liste(s) déployée(s) Si * passé : Référence de l'élément sélectionné

**Note préliminaire :** Elements selectionnes est le nouveau nom de la commande Element selectionne, présente dans les versions précédentes de 4<sup>e</sup> Dimension. Cette commande a été renommée pour mettre en

évidence la possibilité de multi-sélections dans la version 2004 de 4<sup>e</sup> Dimension.

Les capacités de la commande Elements selectionnes ont été étendues dans 4<sup>e</sup> Dimension 2004 :

- la commande peut désormais travailler soit avec les positions courantes des éléments, soit avec leurs références absolues.
- en outre, la commande prend en charge les sélections multiples : elle retourne désormais un tableau contenant les positions ou les références de tous les éléments sélectionnés dans la liste.

Vous pouvez désormais passer dans le paramètre *tabEléments* un tableau d'entiers longs. Si nécessaire, le tableau sera créé et redimensionné par la commande. A l'issue de l'exécution de la commande, *tabEléments* contiendra :

- la position de chaque élément sélectionné parmi la ou les liste(s) déployée(s) si le paramètre \* est omis.
- la référence absolue de chaque élément sélectionné si le paramètre \* est passé.

Le tableau est retourné vide si aucun élément n'est sélectionné.

Le paramètre \* permet d'indiquer si vous souhaitez travailler avec des positions d'éléments (dans ce cas, le paramètre \* doit être omis) ou des références absolues d'éléments (dans ce cas, le paramètre \* doit être passé).

En fonction du paramètre \*, la commande retourne la position ou la référence de l'élément courant de *liste*. En cas de sélection multiple, l'élément courant est le dernier élément sur lequel l'utilisateur a cliqué (sélection manuelle) ou l'élément désigné par la commande **SELECTIONNER ELEMENTS PAR POSITION** ou **SELECTIONNER ELEMENTS PAR REFERENCE** (sélection par programmation).

## Nombre elements

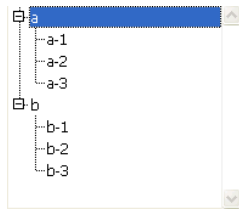
Nombre element(liste{; \*}) → Numérique

Paramètres	Type	Description
liste	RéfListe	→ Numéro de référence de liste
*	*	→ <i>Si omis (défaut) : Retourner les éléments visibles (déployés) de la liste</i> <i>Si spécifié : Retourner tous les éléments de la liste</i>
Résultat	Numérique	← Nombre d'éléments visibles (déployés) dans la liste (si * omis) ou <i>Nombre total d'éléments dans la liste (si * passé)</i>

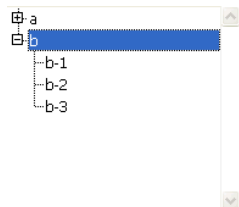
La commande Nombre elements accepte désormais le paramètre facultatif \*. Lorsque ce paramètre est passé, la commande retourne le nombre total d'éléments présents dans la *liste*, quel que soit son état courant déployé/contracté.

Lorsque ce paramètre est omis, la commande retourne le nombre d'éléments déployés, c'est-à-dire visibles, dans la liste.

▼ Soit la liste hiérarchique suivante :



Voici une comparaison des valeurs retournées par la commande Nombre elements lorsque la liste est dans l'état suivant :



```
C_ENTIER LONG($vLnombre;$vLliste)
$vLnombre:=Nombre elements($vLliste) ` $vLnombre vaut 5
$vLnombre:=Nombre elements($vLliste;*) ` $vLnombre vaut 8
```

# Utilisateurs et groupes

Le système de gestion des accès et des mots de passe a été modifié dans 4<sup>e</sup> Dimension 2004. Du point de vue du langage, les principales modifications de la nouvelle version du système sont les suivantes :

- extension des possibilités de paramétrage du système par programmation (accès aux plug-ins, choix du compte utilisateur...);
- possibilité de conserver dans un BLOB crypté l'ensemble des utilisateurs de la base ;
- possibilité d'utiliser une boîte de dialogue personnalisée pour la saisie du nom et du mot de passe afin d'édicter des règles de saisie et/ou d'expiration pour le mot de passe (nombre minimum de caractères, demande de changement périodique...);

Ces modifications se traduisent par de nouvelles commandes et la modification de commandes existantes, décrites dans ce paragraphe.

*Note* Le système de gestion des accès a également bénéficié de modifications d'interface : nouveaux éditeurs d'utilisateurs et de groupes, possibilité pour l'utilisateur de changer son mot de passe "à la volée" et d'enregistrer l'accès à la base. Ces nouveautés sont traitées dans le [paragraphe "Boîte de dialogue d'identification", page 50](#).

## Nouvelles commandes

### ECRIRE ACCES PLUGIN

ECRIRE ACCES PLUGIN (plugIn; groupe)

Paramètres	Type	Description
plugIn	Entier long	→ Numéro du plug-in
groupe	Alpha	→ Nom du groupe à associer au plug-in

La commande ECRIRE ACCES PLUGIN permet de spécifier par programmation le groupe d'utilisateurs autorisé à utiliser chaque plug-in "sérialisé" installé dans la base. Cette définition permet de gérer la répartition des licences des plug-ins.

*Note* Dans les versions précédentes de 4<sup>e</sup> Dimension, cette opération ne pouvait être effectuée qu'en mode Structure via la commande **Répartir les accès aux plug-ins...**

Passez dans le paramètre *plugIn* le numéro du plug-in auquel associer un groupe d'utilisateurs. Les licences de plug-ins incluent les licences Web et SOAP de 4D Client. Vous pouvez passer une des constantes suivantes du thème "Licence disponible" :

Constante	Type	Valeur
Licence 4D Draw	Entier long	808464694
Licence 4D for OCI	Entier long	808465208
Licence 4D View	Entier long	808465207
Licence 4D Write	Entier long	808464697
Licence Web 4D Client	Entier long	808465209
Licence SOAP 4D Client	Entier long	808465465
Licence 4D ODBC Pro	Entier long	808464946

Passez dans le paramètre *groupe* le nom du groupe dont les utilisateurs seront autorisés à utiliser le plug-in.

**Référence :** Lire acces plugin

### Lire acces plugin

Lire acces plugin (plugIn) → Alpha

Paramètres	Type	Description
plugIn	Entier long	→ Numéro du plug-in
Résultat	Alpha	← Nom du groupe associé au plug-in

La commande Lire acces plugin retourne le nom du groupe d'utilisateurs autorisé à utiliser le plug-in dont le numéro a été passé dans le paramètre *plugIn*. Si aucun groupe n'est associé au plug-in, la commande retourne une chaîne vide ("").

Passez dans le paramètre *plugIn* le numéro du plug-in duquel vous souhaitez connaître le groupe d'utilisateurs associé. Les licences de plug-ins incluent les licences Web et SOAP de 4D Client. Vous pouvez passer une des constantes du thème "Licence disponible".

**Référence :** ECRIRE ACCES PLUGIN



**LIRE LISTE PLUGIN**

LIRE LISTE PLUGIN (tabNuméros; tabNoms)

Paramètres	Type	Description
tabNuméros	Tab Entier long	→ Numéros des plug-ins
tabNoms	Tab Alpha	→ Noms des plug-ins

La commande LIRE LISTE PLUGIN remplit les tableaux *tabNuméros* et *tabNoms* avec les numéros et les noms des plug-ins chargés par l'application 4e Dimension. Les deux tableaux sont automatiquement dimensionnés et synchronisés par la commande.

*Note* Vous pouvez comparer les valeurs retournées dans le tableau *tabNuméros* avec les constantes du thème "Licence disponible".

LIRE LISTE PLUGIN prend en compte tous les plug-ins, y compris les plug-ins intégrés (par exemple 4D Chart) et les plug-ins des éditeurs tiers.

**UTILISATEURS VERS BLOB**

UTILISATEURS VERS BLOB (utilisateurs)

Paramètres	Type	Description
utilisateurs	BLOB	→ BLOB devant contenir les utilisateurs ← Comptes utilisateurs (crypté)

La commande UTILISATEURS VERS BLOB stocke dans le BLOB *utilisateurs* la liste de tous les comptes d'utilisateurs et les groupes de la base créés par l'Administrateur.

Seul l'Administrateur de la base peut exécuter cette commande. Si un autre utilisateur tente de l'exécuter, la commande ne fait rien et une erreur de privilège (-9949) est générée.

Le BLOB généré est automatiquement encrypté et ne peut être lu que par la commande **BLOB VERS UTILISATEURS**. Vous pouvez stocker ce BLOB dans un fichier disque ou dans un champ.

Cette commande est semblable à l'enregistrement des groupes et utilisateurs depuis la fenêtre de gestion des groupes de la Boîte à outils, à la différence près qu'elle permet de stocker les comptes utilisateurs dans un champ BLOB et non uniquement dans un fichier.

Ce principe permet de conserver une sauvegarde des utilisateurs parmi les données de la base, et ainsi de mettre en place un mécanisme de sauvegarde et de chargement automatiques des utilisateurs en cas de mise à jour de la structure de la base (en effet, les informations relatives aux comptes utilisateurs sont stockées par 4<sup>e</sup> Dimension dans le fichier de structure de la base).

**Référence : BLOB VERS UTILISATEURS**

**BLOB VERS  
UTILISATEURS**

BLOB VERS UTILISATEURS (utilisateurs)

Paramètres	Type	Description
utilisateurs	BLOB	→ BLOB (crypté) contenant des comptes utilisateurs créés et sauvegardés par l'Administrateur

La commande BLOB VERS UTILISATEURS ajoute dans la base de données les comptes utilisateurs présents dans le BLOB *utilisateurs*. Le BLOB *utilisateurs* est crypté et doit impérativement avoir été créé par la commande **UTILISATEURS VERS BLOB**.

Seul l'Administrateur de la base peut exécuter cette commande. Si un autre utilisateur tente de l'exécuter, la commande ne fait rien et une erreur de privilège (-9949) est générée.

L'ajout de chaque compte utilisateur est effectué en tenant compte des règles suivantes :

- Le numéro d'identification de l'utilisateur sert de référence. Les utilisateurs sont traités dans l'ordre des numéros.
- Si le numéro existe déjà dans la structure de la base, les informations du compte sont mises à jour si nécessaire en fonction des informations contenues dans le BLOB.
- Si le numéro n'existe pas dans la structure de la base, l'utilisateur est créé en fonction des informations contenues dans le BLOB.
- Si le numéro correspond à un compte utilisateur supprimé dans la structure de la base, le compte est mis à jour en fonction des informations contenues dans le BLOB.
- Si les informations contenues dans le BLOB indiquent que le compte utilisateur est supprimé, le compte est supprimé dans la structure de la base.

- Les utilisateurs mis à jour sont associés aux groupes suivant les informations du BLOB.
- Si un groupe n'existe pas, il est ajouté.

Si la commande est exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0.

**Note de compatibilité :** Les fichiers d'utilisateurs et groupes (extension .4UG) créés par la commande de menu **Enregistrer les groupes & utilisateurs...** dans 4<sup>e</sup> Dimension 2003 peuvent être chargés dans 4<sup>e</sup> Dimension 2004 via la séquence suivante :

**DOCUMENT VERS BLOB**(mondoc; blob)  
**BLOB VERS UTILISATEURS**(blob)

En revanche, les fichiers d'utilisateurs et groupes générés dans 4<sup>e</sup> Dimension 2004 ne peuvent pas être ouverts en version 2003.

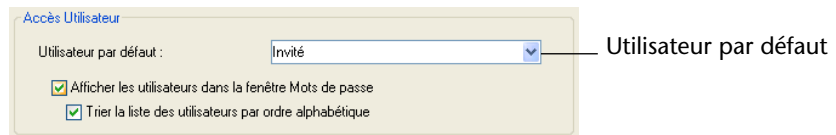
*Référence :* **UTILISATEURS VERS BLOB**

## Lire utilisateur par défaut

Lire utilisateur par défaut → Numérique

Paramètres	Type	Description
Résultat	Numérique	← Numéro de référence unique de l'utilisateur

Cette commande retourne le numéro de référence unique de l'utilisateur désigné comme "Utilisateur par défaut" dans la boîte de dialogue des Préférences de la base :



Les numéros de référence pour les utilisateurs peuvent être les suivants :

Numéro de référence	Description de l'utilisateur
1	Super_Utilisateur
2	Administrateur

3 à 15000	Utilisateur créé par le Super_Utilisateur (l'utilisateur n°3 est le 1er utilisateur créé par le Super_Utilisateur, l'utilisateur n°4 est le deuxième, et ainsi de suite).
-11 à -15010	Utilisateur créé par l'Administrateur (l'utilisateur n°-11 est le 1er utilisateur créé par l'Administrateur, l'utilisateur n°-12 est le deuxième, et ainsi de suite).

Si aucun utilisateur par défaut n'est défini, la commande retourne 0.

## Commandes modifiées

### CHANGER UTILISATEUR COURANT (CHANGER UTILISATEUR)

CHANGER UTILISATEUR COURANT  $\{(nom \mid réfUtilisateur; motDePasse)\}$

Paramètres	Type	Description
$nom \mid réfUtilisateur$	Alpha   Num	→ Nom ou Numéro de référence unique de l'utilisateur
$motDePasse$	Alpha	→ Mot de passe (non crypté)

**Note préliminaire** : CHANGER UTILISATEUR COURANT est le nouveau nom de la commande CHANGER UTILISATEUR, déjà présente dans les versions précédentes de 4<sup>e</sup> Dimension. Cette commande a été renommée pour plus de clarté.

La commande CHANGER UTILISATEUR COURANT accepte désormais deux paramètres facultatifs, permettant de spécifier par programmation le nouveau compte à utiliser pour tous les process "utilisateurs" courants. A noter qu'un process créé par un autre process hérite du compte utilisateur de celui-ci.

---

*4D Server* Sur le poste serveur, les process démarrés via les méthodes bases Sur démarrage serveur, Sur arrêt serveur, etc., sont exécutés sous le compte du Super\_Utilisateur.

---

Passez comme premier paramètre le *nom* ou le numéro de référence unique (*réfUtilisateur*) du compte à utiliser. Les noms et les numéros des utilisateurs peuvent être obtenus via la commande LIRE LISTE UTILISATEURS.

Les numéros de référence pour les utilisateurs peuvent être les suivants :

Numéro de référence	Description de l'utilisateur
1	Super_Utilisateur
2	Administrateur
3 à 15000	Utilisateur créé par le Super_Utilisateur (l'utilisateur n°3 est le 1er utilisateur créé par le Super_Utilisateur, l'utilisateur n°4 est le deuxième, et ainsi de suite).
-11 à -15010	Utilisateur créé par l'Administrateur (l'utilisateur n°-11 est le 1er utilisateur créé par l'Administrateur, l'utilisateur n°-12 est le deuxième, et ainsi de suite).

Si le compte d'utilisateur n'existe pas ou a été supprimé, l'erreur -9979 est générée. Vous pouvez intercepter cette erreur avec une méthode de gestion d'erreurs installée par la commande APPELER SUR ERREUR. Sinon, vous pouvez appeler la fonction Utilisateur supprime pour tester le compte utilisateur avant d'appeler cette commande.

Passez dans le paramètre *motDePasse* le mot de passe non crypté du compte de l'utilisateur. Si le mot de passe ne correspond pas à l'utilisateur, la commande ne fait rien et l'erreur -9978 est générée.

La commande est désormais temporisée afin d'éviter des attaques par "force brute" (essais automatiques de multiples combinaisons de noms d'utilisateurs/mots de passe). Ainsi, au bout du quatrième appel à la commande, elle n'est exécutée qu'après 10 secondes d'attente. Cette temporisation est globale au poste de travail.

---

*Note* La commande Valider mot de passe bénéficie également désormais d'une temporisation de 10 secondes au bout de quatre appels infructueux.

---

## LIRE PROPRIETES UTILISATEUR

LIRE PROPRIETES UTILISATEUR (réfUtilisateur; nom; démarrage; motDePasse; nbUtilisations; dernièreUtilisation{; adhésions{; *groupePropriétaire*})

Paramètres	Type	Description
réfUtilisateur	Numérique	→ Numéro de référence unique de l'utilisateur
nom	Alpha	← Nom de l'utilisateur
démarrage	Alpha	← Nom de la méthode de démarrage
motDePasse	Alpha	← Chaîne vide
nbUtilisations	Numérique	← Nombre d'utilisations de la base
dernièreUtilisation	Date	← Date de la dernière utilisation de la base
adhésions	Tableau Num	← Numéros de référence des groupes auxquels l'utilisateur appartient
<i>groupePropriétaire</i>	<i>Numérique</i>	← <i>Numéro de référence du groupe propriétaire de l'utilisateur</i>

La commande LIRE PROPRIETES UTILISATEUR retourne désormais dans le paramètre facultatif *groupePropriétaire* le numéro de référence du groupe "propriétaire" de l'utilisateur, c'est-à-dire le groupe propriétaire par défaut des objets créés par cet utilisateur.

## Ecrire proprietes utilisateur

Ecrire proprietes utilisateur (réfUtilisateur; nom; démarrage; motDePasse; nbUtilisations; dernièreUtilisation{; adhésions{; *groupePropriétaire*}) → Numérique

Paramètres	Type	Description
réfUtilisateur	Numérique	→ Numéro de référence unique de l'utilisateur (ou -1 ou -2)
nom	Alpha	→ Nouveau nom de l'utilisateur
démarrage	Alpha	→ Nom de la méthode de démarrage
motDePasse	Alpha	→ Mot de passe ou *
nbUtilisations	Numérique	→ Nombre d'utilisations de la base
dernièreUtilisation	Date	→ Date de la dernière utilisation de la base
adhésions	Tableau Num	→ Numéros de référence des groupes auxquels l'utilisateur appartient
<i>groupePropriétaire</i>	<i>Numérique</i>	→ <i>Numéro de référence du groupe propriétaire de l'utilisateur</i>
Résultat	Numérique	← Numéro de référence unique du nouvel utilisateur

La commande Ecrire proprietes utilisateur permet désormais de modifier, à l'aide du paramètre facultatif *groupePropriétaire*, le numéro de référence du groupe "propriétaire" de l'utilisateur, c'est-à-dire le groupe propriétaire par défaut des objets créés par cet utilisateur.

## Proposer une boîte de dialogue de gestion d'accès personnalisée

Les nouveautés du thème "Utilisateurs et groupes" permettent de mettre en place des boîtes de dialogue personnalisées pour la saisie du nom et du mot de passe tout en bénéficiant des avantages du système intégré de contrôle des accès de 4<sup>e</sup> Dimension.

Il est désormais possible de mettre en oeuvre des fonctions avancées de contrôle des mots de passe telles que :

- édicter des règles de saisie (nombre minimum de caractères, unicité...)
- établir des règles d'expiration (date d'expiration, changement forcé à la première connexion, verrouillage du compte après plusieurs saisies erronées, mémorisation des mots de passe déjà utilisés...).

Le principe est le suivant :

- 1 **L'entrée dans la base s'effectue directement en mode "Utilisateur par défaut", sans boîte de dialogue.**
- 2 **Dans la méthode base Sur ouverture, le développeur provoque l'affichage d'une boîte de dialogue personnalisée de saisie du nom d'utilisateur et du mot de passe (à l'aide de la commande DIALOGUE ou AJOUTER ENREGISTREMENT par exemple).**

Ce mécanisme permet d'envisager tout type de traitement :

- Il est possible d'afficher la liste des utilisateurs de la base, comme dans la boîte de dialogue d'accès standard de 4<sup>e</sup> Dimension, à l'aide de la commande LIRE LISTE UTILISATEURS.
  - Le champ de saisie du mot de passe peut contenir divers contrôles afin de vérifier la validité des caractères saisis.
  - Les règles d'expiration peuvent être appliquées au moment de la validation de la boîte de dialogue : première connexion, date d'expiration, verrouillage du compte en cas de saisies erronées répétées, etc.
- 3 **Lorsque la saisie est validée, les informations requises (nom d'utilisateur et mot de passe) sont passées à la commande CHANGER UTILISATEUR COURANT afin d'ouvrir la base avec les privilèges du compte utilisateur.**

## Fenêtres

Le thème de commandes “Fenêtres” a été enrichi dans 4<sup>e</sup> Dimension 2004 afin de fournir aux développeurs davantage de moyens de contrôle sur le dimensionnement des fenêtres. De nouvelles commandes ont été ajoutées ([Fenetre formulaire courant](#) et [REDIMENSIONNER FENETRE FORMULAIRE](#)) et le comportement de la commande existante [CHANGER COORDONNEES FENETRE](#) a été modifié.

A noter également que la gestion par défaut du redimensionnement des fenêtres a été modifiée ; pour plus d’informations sur ce point, reportez-vous au [paragraphe “Bords pousseurs”](#), page 283.

Enfin, deux nouveaux types de fenêtres sont disponibles :

- Fenêtre pop up (type 32) pour la commande Créer fenetre.
- Fenêtre feuille (type 33) pour les commandes Créer fenetre et Créer fenetre formulaire.

### Nouvelles commandes

#### Fenetre formulaire courant

Fenetre formulaire courant → RefFen

Paramètres	Type	Description
		Cette commande ne requiert pas de paramètre

Résultat	RefFen	← Numéro de référence de la fenêtre du formulaire courant
----------	--------	---

Cette commande retourne la référence de la fenêtre du formulaire courant. S’il n’y a pas de fenêtre définie pour le formulaire courant, la commande retourne 0.

La fenêtre du formulaire courant peut avoir été générée automatiquement par une commande telle que AJOUTER ENREGISTREMENT, à la suite d’une action utilisateur ou via les commandes Créer fenetre ou Créer fenetre formulaire.



## REDIMENSIONNER FENETRE FORMULAIRE

REDIMENSIONNER FENETRE FORMULAIRE(largeur; hauteur)

Paramètres	Type	Description
largeur	Entier long	→ Pixels à ajouter ou soustraire à la largeur courante de la fenêtre formulaire
hauteur	Entier long	→ Pixels à ajouter ou soustraire à la hauteur courante de la fenêtre formulaire

Cette commande permet de modifier la taille de la fenêtre du formulaire courant.

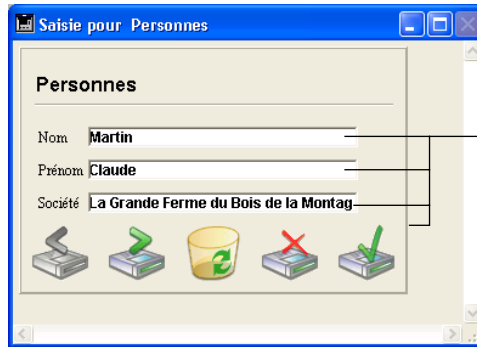
Passez dans les paramètres *largeur* et *hauteur* le nombre de pixels que vous souhaitez ajouter aux dimensions courantes de la fenêtre. Pour ne pas modifier une dimension, passez 0 dans le paramètre correspondant. Pour réduire une dimension, passez une valeur négative dans *largeur* et *hauteur*.

Cette commande produit exactement le même résultat qu'un redimensionnement manuel de la fenêtre à l'aide de la case de redimensionnement (si le type de fenêtre le permet). Par conséquent, la commande tient compte des propriétés de redimensionnement des objets et des contraintes de taille définies dans les propriétés du formulaire : si par exemple la commande entraîne un redimensionnement de la fenêtre supérieur à la taille maximale du formulaire, elle n'a pas d'effet.

A noter que ce fonctionnement est différent de celui de la commande **CHANGER COORDONNEES FENETRE**, qui ne tient pas compte des propriétés du formulaire ni de son contenu en cas de redimensionnement de la fenêtre. Pour plus d'informations sur les modifications apportées à cette commande dans 4<sup>e</sup> Dimension 2004, reportez-vous à la description de la [commande CHANGER COORDONNEES FENETRE](#), page 282.

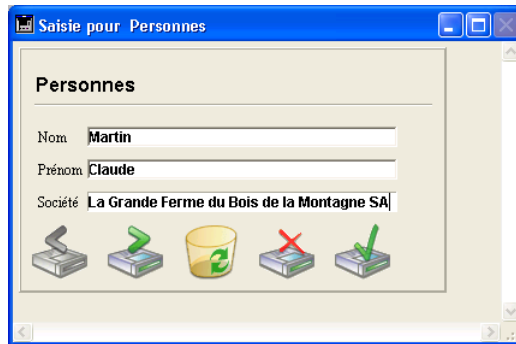
A noter également que cette commande ne modifie pas forcément les dimensions du formulaire lui-même. Pour modifier par programmation la taille d'un formulaire, reportez-vous à la description de la [commande FIXER TAILLE FORMULAIRE](#), page 289.

- ▼ Cet exemple illustre l'action de la commande :



Ces objets ont pour propriété de redimensionnement horizontal "Agrandir"

### REDIMENSIONNER FENETRE FORMULAIRE(25;0)



## Commande modifiée

### CHANGER COORDONNEES FENETRE

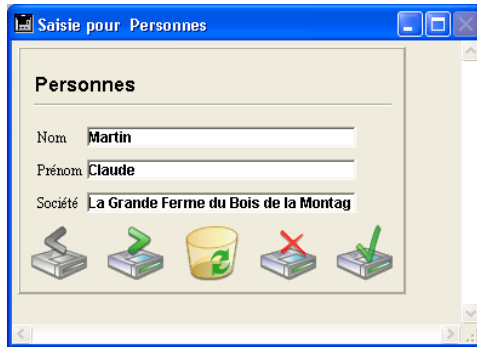
CHANGER COORDONNEES FENETRE (gauche; haut; droite; bas; fenêtre)

Le fonctionnement de cette commande a été modifié dans 4<sup>e</sup> Dimension 2004 : désormais, cette commande n'affecte plus les objets du formulaire.

Dans les versions précédentes du programme, lorsque la fenêtre contenait un formulaire, les objets du formulaire étaient déplacés ou redimensionnés par cette commande (en fonction de leurs propriétés). Désormais, la commande ne tient pas compte des propriétés du formulaire ni de ses objets ; seule la fenêtre est modifiée.

Pour modifier une fenêtre de formulaire avec ses propriétés de redimensionnement et les objets qu'elle contient, vous devez utiliser la nouvelle [commande REDIMENSIONNER FENETRE FORMULAIRE](#), page 281.

- ▼ Cet exemple illustre l'action de la commande :



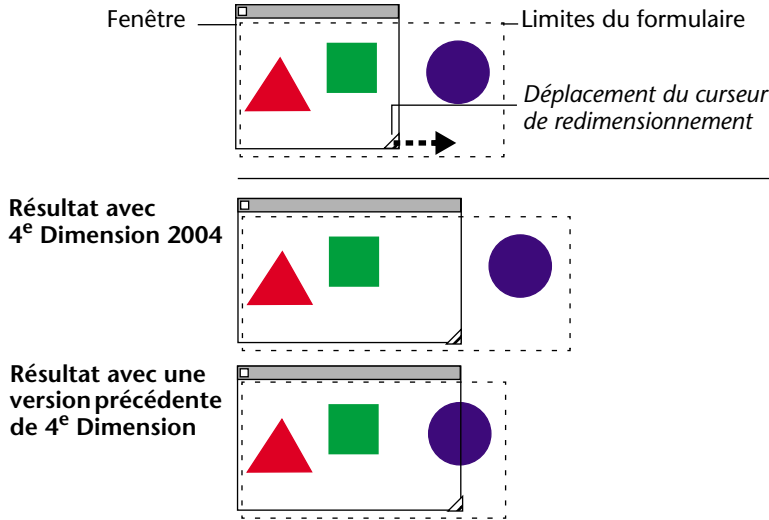
CHANGER COORDONNEES FENETRE (100;100;300;300)



## Bords pousseurs

Dans 4<sup>e</sup> Dimension 2004, les bords droits et bas des fenêtres sont devenus par défaut des séparateurs “pousseurs”. Cela signifie que les objets se trouvant à droite ou au-dessous des limites d’une fenêtre

affichée à l'écran seront automatiquement repoussés vers la droite ou vers le bas en cas d'agrandissement de la fenêtre :

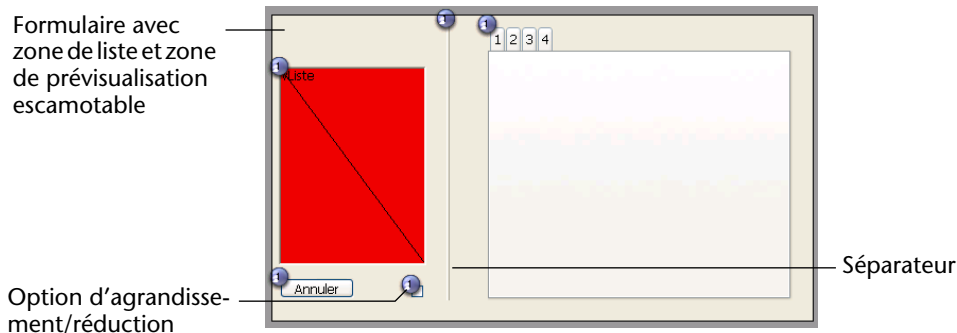


*Note* Pour plus d'informations sur les séparateurs "pousseurs", reportez-vous au [paragraphe "Gestion des séparateurs"](#), page 146.

Combiné aux nouveautés liées à la gestion des fenêtres, ce mécanisme permet de gérer des fenêtres à volets escamotables de type Explorateur (cf. exemple ci-dessous).

Ce principe n'est pas mis en oeuvre lorsque la fenêtre comporte des barres de défilement.

- ▼ Voici un exemple de mise en place d'une fenêtre de type Explorateur. Le formulaire suivant est défini en mode Structure :



*Note* La taille du formulaire est "automatique".

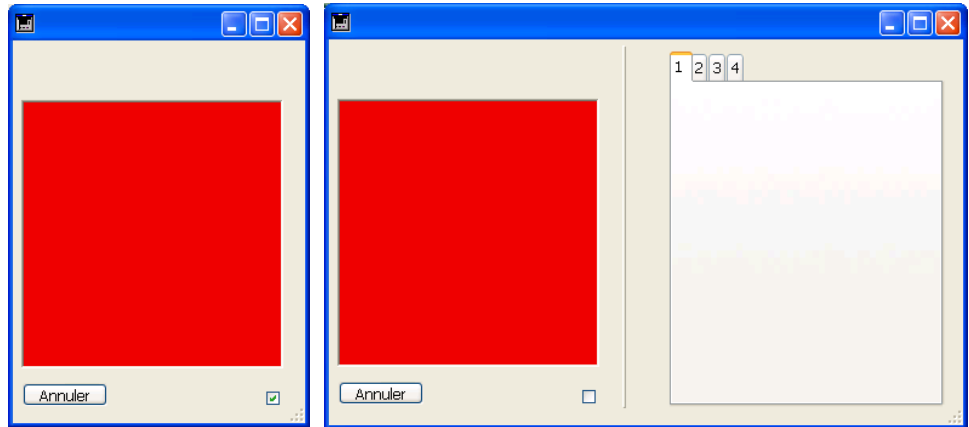
La fenêtre est affichée via l'instruction suivante :

```
$ref:=Creer fenetre formulaire([Table 1];"Form1";Form fenêtre standard ;  
Centrée horizontalement ;Centrée verticalement ;*)
```

```
DIALOGUE([Table 1];"Form1")
```

**FERMER FENETRE**

La partie droite de la fenêtre peut être affichée ou masquée via un clic sur l'option d'agrandissement/réduction :



La méthode objet associée à cette option est la suivante :

**Au cas ou**

```
: (Evenement formulaire=Sur chargement )
```

```
  C_BOOLEEN(b1;<>contracté)
```

```
  C_ENTIER LONG(marge)
```

```
  marge:=15
```

```
  b1:=<>contracté
```

```
  Si (<>contracté)
```

```
    FIXER REDIMENSIONNEMENT FORMULAIRE HORIZONTAL(Faux)
```

```
    FIXER TAILLE FORMULAIRE("b1";marge;marge)
```

```
  Sinon
```

```
    FIXER REDIMENSIONNEMENT FORMULAIRE HORIZONTAL(Vrai)
```

```
    FIXER TAILLE FORMULAIRE("onglet";marge;marge)
```

```
  Fin de si
```

```
: (Evenement formulaire=Sur clic )
```

```
  <>contracté:=b1
```

Si (b1)

\contracté

LIRE RECT OBJET(\*;"b1";\$g;\$h;\$d;\$b)

COORDONNEES FENETRE(\$gf;\$hf;\$df;\$bf;Fenetre formulaire  
courant)

CHANGER COORDONNEES FENETRE(\$gf;\$hf;\$gf+\$d+marge;  
\$hf+\$b+marge;Fenetre formulaire courant)

FIXER REDIMENSIONNEMENT FORMULAIRE HORIZONTAL(Faux)

FIXER TAILLE FORMULAIRE("b1";marge;marge)

Sinon

\déployé

LIRE RECT OBJET(\*;"onglet";\$g;\$h;\$d;\$b)

COORDONNEES FENETRE(\$gf;\$hf;\$df;\$bf;Fenetre formulaire  
courant)

CHANGER COORDONNEES FENETRE(\$gf;\$hf;\$gf+\$d+marge;  
\$hf+\$b+marge;Fenetre formulaire courant)

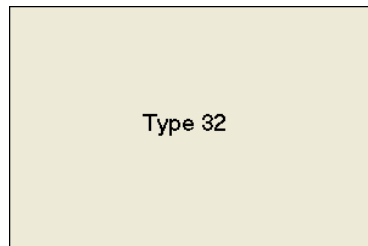
FIXER REDIMENSIONNEMENT FORMULAIRE HORIZONTAL(Vrai)

FIXER TAILLE FORMULAIRE("onglet";marge;marge)

Fin de si

Fin de cas

**Type Fenêtre pop up** Une nouvelle constante a été ajoutée dans les thèmes “Creer fenetre” et “Creer fenetre formulaire” : Fenêtre pop up (Form fenetre pop up) de valeur 32. L'apparence des fenêtres de type 32 est la suivante :



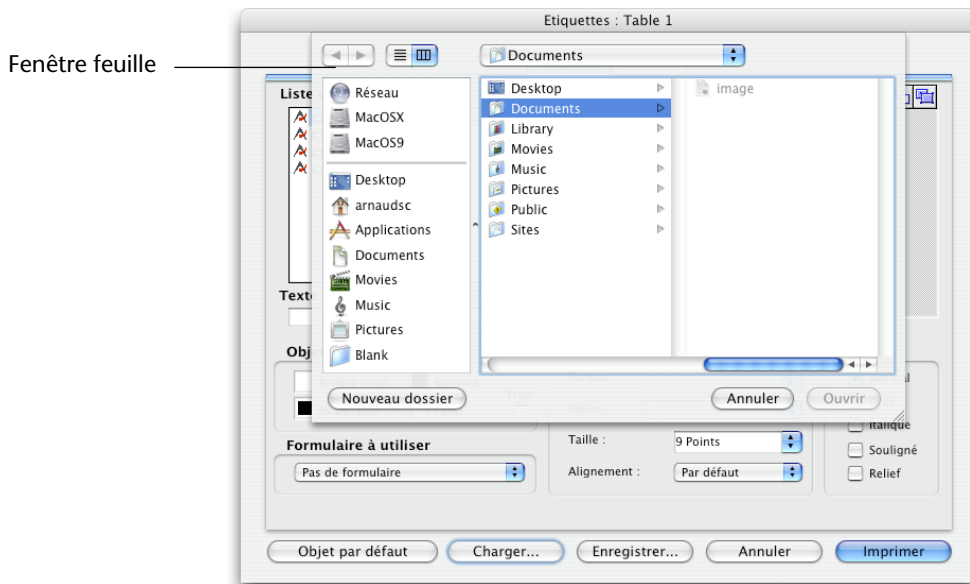
Ce type de fenêtre reprend les caractéristiques élémentaires des fenêtres de type 2 (Dialogue simple) et dispose de propriétés avancées spécifiques :

- La fenêtre est automatiquement refermée avec annulation lorsque :
  - un clic se produit en-dehors de la fenêtre ;
  - la fenêtre d'arrière-plan ou la fenêtre MDI est déplacée ;
  - l'utilisateur appuie sur la touche **Echap** (ou **Esc**).

- Cette fenêtre s’affiche devant sa fenêtre “parente” (elle ne doit pas être utilisée comme fenêtre principale d’un process). La fenêtre d’arrière-plan n’est pas désactivée. En revanche, elle ne reçoit plus d’événement.
- Il n’est pas possible de redimensionner ou de déplacer la fenêtre à l’aide de la souris ; toutefois, lorsque cette opération est effectuée par programmation, le redessinement des éléments d’arrière-plan est optimisé.

Ce type de fenêtre est principalement destiné à la prise en charge des pop up menus associés aux boutons 3D de type “bevel” ou “barres outils” (cf. [paragraphe “Modification des boutons 3D”, page 155](#)).

**Type Fenêtre feuille** Les fenêtres feuilles (*sheet windows*) sont des fenêtres spécifiques de l’interface MacOS X. Ces fenêtres “descendent” de la barre de titre de la fenêtre principale via une animation et s’affichent par-dessus celle-ci. Leurs propriétés sont identiques à celles des boîtes de dialogue modales. Elles sont généralement utilisées pour effectuer une action en relation directe avec celle se déroulant dans la fenêtre principale. Dans 4<sup>e</sup> Dimension par exemple, une fenêtre feuille est utilisée lorsque l’utilisateur clique sur le bouton **Charger** dans l’éditeur d’étiquettes :



Il n’est possible de créer une fenêtre feuille sous Mac OS X que si la dernière fenêtre ouverte est visible et de type document (formulaire).

De nouvelles constantes ont été ajoutées dans 4<sup>e</sup> Dimension afin de gérer les fenêtres feuille :

- dans le thème “Creer fenetre” : Fenêtre feuille (33) et Fenêtre feuille redim (34).
- dans le thème “Creer fenetre formulaire” : Form fenêtre feuille (33).

### Creer fenetre

Une fenêtre feuille est créée lorsque vous passez le type Fenêtre feuille (33) à la commande Creer fenetre. La fenêtre feuille est redimensionnable si vous passez le type Fenêtre feuille redim (34).

La commande crée une fenêtre de type 1 au lieu du type 33 ou 8 au lieu du type 34 :

- si la dernière fenêtre ouverte n'est pas visible ou n'est pas de type document,
- sous Windows.

### Creer fenetre formulaire

Une fenêtre feuille est créée lorsque vous passez le type Form fenêtre feuille (33) à la commande Creer fenetre formulaire. La fenêtre est redimensionnable si le formulaire l'est.

La commande crée une fenêtre de type 1 (ou de type 8 si le formulaire est redimensionnable) :

- si la dernière fenêtre ouverte n'est pas visible ou n'est pas de type document,
  - sous Windows.
- ▼ Comme une fenêtre feuille doit être dessinée par-dessus un formulaire, son affichage est repoussé dans l'événement Sur chargement du premier formulaire chargé dans la fenêtre (cf exemple). Cet exemple illustre ce “retard” d'affichage :

```
$maFenetre:=Creer fenetre formulaire([Table];"formulaire";  
                                         Form fenêtre feuille)  
`La fenêtre formulaire est créée mais reste invisible  
DIALOGUE([Table];"formDial")  
`L'événement Sur chargement est généré puis la fenêtre feuille est  
`affichée
```



## Formulaires

Le nouveau thème "Formulaires" regroupe les commandes de contrôle par programmation des formulaires. Il contient :

- les nouvelles commandes 2004 de gestion des formulaires : **FIXER TAILLE FORMULAIRE**, **FIXER REDIMENSIONNEMENT FORMULAIRE HORIZONTAL**, **FIXER REDIMENSIONNEMENT FORMULAIRE VERTICAL** et **LIRE OBJETS FORMULAIRE**,
- les commandes modifiées **FORMULAIRE ENTREE** et **FORMULAIRE SORTIE**, auparavant situées dans le thème "Table",
- la commande existante **LIRE PROPRIETES FORMULAIRE**, auparavant située dans le thème "Fenêtres" (son fonctionnement est inchangé),
- les commandes auparavant situées dans le thème "Pages formulaire" : **PREMIERE PAGE**, **DERNIERE PAGE**, **PAGE SUIVANTE**, **PAGE PRECEDENTE**, **ALLER A PAGE** et **Page formulaire courante**. Le thème "Pages formulaire" n'existe plus dans 4<sup>e</sup> Dimension 2004.

### FIXER TAILLE FORMULAIRE

**FIXER TAILLE FORMULAIRE**({objet; }horizontal; vertical{\* })

Paramètres	Type	Description
objet	Alpha	→ Nom d'objet indiquant les limites du formulaire
horizontal	Entier long	→ Si * passé : marge horizontale (pixels) Si * omis : largeur (pixels)
vertical	Entier long	→ Si * passé : marge verticale (pixels) Si * omis : hauteur (pixels)
*	*	→ Si passé : - taille automatique, ajouter les marges définies par les paramètres horizontal et vertical, ou - taille basée sur objet (si objet passé), ajouter les marges définies par les paramètres horizontal et vertical Si omis : - utiliser horizontal et vertical comme largeur et hauteur du formulaire

Cette commande permet de modifier par programmation la taille du formulaire courant. La nouvelle taille est définie pour le process courant, elle n'est pas stockée avec le formulaire.

Comme en mode Structure, cette commande permet de définir la taille d'un formulaire de trois manières :

- automatiquement — 4<sup>e</sup> Dimension détermine la taille du formulaire sur le principe que tous les objets doivent être visibles — en ajoutant éventuellement une marge horizontale et une marge verticale,
- sur la base de l'emplacement d'un objet du formulaire auquel s'ajoutent éventuellement une marge horizontale et une marge verticale,
- en saisissant des dimensions "absolues" (largeur et hauteur).

Pour plus d'informations sur les possibilités de dimensionnement des formulaires, reportez-vous au manuel *Mode Structure* de 4<sup>e</sup> Dimension.

### Taille automatique

Pour que le formulaire ait une taille automatique, vous devez utiliser la syntaxe suivante :

```
FIXER TAILLE FORMULAIRE(horizontal; vertical;*)
```

Dans ce cas, vous devez passer dans *horizontal* et *vertical* les marges (en pixels) que vous souhaitez ajouter à droite et en bas du formulaire.

### Taille basée sur un objet

Pour que la taille du formulaire soit basée sur un objet, vous devez utiliser la syntaxe suivante :

```
FIXER TAILLE FORMULAIRE(objet; horizontal; vertical;*)
```

Dans ce cas, vous devez passer dans *horizontal* et *vertical* les marges (en pixels) que vous souhaitez ajouter à droite et en bas de l'objet.

### Taille en valeur absolue

Pour passer une taille de formulaire absolue, vous devez utiliser la syntaxe suivante :

```
FIXER TAILLE FORMULAIRE(horizontal; vertical)
```

Dans ce cas, vous devez passer dans *horizontal* et *vertical* la largeur et la hauteur (en pixels) du formulaire.

La commande `FIXER TAILLE FORMULAIRE` modifie la taille du formulaire mais tient compte de ses propriétés de redimensionnement. Par exemple, si la largeur minimale du formulaire est de 500 pixels et si la commande définit une largeur de 400 pixels, la nouvelle largeur du formulaire sera de 500 pixels.

A noter également que cette commande ne modifie pas la taille de la fenêtre du formulaire (il est possible de redimensionner un formulaire sans que la taille de la fenêtre soit modifiée, et inversement). Pour modifier la taille de la fenêtre d'un formulaire, reportez-vous à la description de la [commande REDIMENSIONNER FENETRE FORMULAIRE](#), page 281.

**FIXER  
REDIMENSIONNEMENT  
FORMULAIRE  
HORIZONTAL**

FIXER REDIMENSIONNEMENT FORMULAIRE HORIZONTAL(redimension{  
largeurMini{; largeurMaxi}})

Paramètres	Type	Description
redimension	Booléen	→ Vrai : le formulaire est redimensionnable horizontalement Faux : le formulaire n'est pas redimensionnable horizontalement
largeurMini	Entier long	→ Largeur minimale du formulaire (pixels)
largeurMaxi	Entier long	→ Largeur maximale du formulaire (pixels)

La commande **FIXER REDIMENSIONNEMENT FORMULAIRE HORIZONTAL** permet de modifier par programmation les propriétés de redimensionnement horizontal du formulaire courant. Par défaut, ces propriétés sont définies dans l'éditeur de formulaires en mode Structure. Les nouvelles propriétés sont fixées pour le process courant, elles ne sont pas stockées avec le formulaire.

Le paramètre *redimension* permet de définir si le formulaire est redimensionnable horizontalement, c'est-à-dire si sa largeur est modifiable (manuellement par l'utilisateur ou par programmation). Si vous passez Vrai, la largeur du formulaire peut être modifiée par l'utilisateur ; 4<sup>e</sup> Dimension utilise comme bornes les valeurs éventuellement passées dans les paramètres *largeurMini* et *largeurMaxi*. Si vous passez Faux, la largeur courante du formulaire n'est pas modifiable ; dans ce cas, il est inutile de passer des valeurs dans les paramètres *largeurMini* et *largeurMaxi*.

Si vous avez passé Vrai dans le premier paramètre, vous pouvez passer dans les paramètres facultatifs *largeurMini* et *largeurMaxi* les nouvelles largeurs minimale et maximale du formulaire en pixels. Si vous omettez ces paramètres, les valeurs définies en mode Structure (le cas échéant) seront utilisées.

**FIXER  
REDIMENSIONNEMENT  
FORMULAIRE VERTICAL**

FIXER REDIMENSIONNEMENT FORMULAIRE VERTICAL(redimension{; hauteurMini{; hauteurMaxi{}})

Paramètres	Type	Description
redimension	Booléen	→ Vrai : le formulaire est redimensionnable verticalement Faux : le formulaire n'est pas redimensionnable verticalement
hauteurMini	Entier long	→ Hauteur minimale du formulaire (pixels)
hauteurMaxi	Entier long	→ Hauteur maximale du formulaire (pixels)

La commande FIXER REDIMENSIONNEMENT FORMULAIRE HORIZONTAL permet de modifier par programmation les propriétés de redimensionnement vertical du formulaire courant. Par défaut, ces propriétés sont définies dans l'éditeur de formulaires en mode Structure. Les nouvelles propriétés sont fixées pour le process courant, elles ne sont pas stockées avec le formulaire.

Le paramètre *redimension* permet de définir si le formulaire est redimensionnable verticalement, c'est-à-dire si sa hauteur est modifiable (manuellement par l'utilisateur ou par programmation). Si vous passez Vrai, la hauteur du formulaire peut être modifiée par l'utilisateur ; 4<sup>e</sup> Dimension utilise comme bornes les valeurs éventuellement passées dans les paramètres *hauteurMini* et *hauteurMaxi*.

Si vous passez Faux, la hauteur courante du formulaire n'est pas modifiable ; dans ce cas, il est inutile de passer des valeurs dans les paramètres *hauteurMini* et *hauteurMaxi*.

Si vous avez passé Vrai dans le premier paramètre, vous pouvez passer dans les paramètres facultatifs *hauteurMini* et *hauteurMaxi* les nouvelles hauteurs minimale et maximale du formulaire en pixels. Si vous omettez ces paramètres, les valeurs définies en mode Structure (le cas échéant) seront utilisées.

**LIRE OBJETS  
FORMULAIRE**

LIRE OBJETS FORMULAIRE(tabObjets{; tabVariables{; tabPages{}}{; \*})

Paramètres	Type	Description
tabObjets	Tableau alpha	← Noms des objets du formulaire
tabVariables	Tableau pointeur	← Pointeurs sur les variables ou champs associés aux objets
tabPages	Tableau entier	← Numéro de page de chaque objet
*	*	→ Si passé = réduire à la page courante

La commande LIRE OBJETS FORMULAIRE retourne sous forme de tableau(x) la liste de tous les objets présents dans le formulaire courant de la table courante. Cette liste peut être restreinte à la page courante du formulaire. La commande peut être utilisée avec les formulaires entrée et sortie.

Si un tableau passé en paramètre n'est pas préalablement déclaré, la commande le crée et le dimensionne automatiquement. Toutefois, dans la perspective de la compilation de l'application, il est recommandé de déclarer explicitement chaque tableau.

Passez dans *tabObjets* le nom du tableau alpha devant être rempli avec les noms des objets (chaque nom d'objet est unique au sein d'un formulaire). L'ordre dans lequel les objets apparaissent dans le tableau n'est pas significatif.

Les autres tableaux remplis facultativement par la commande sont synchronisés avec le premier.

Passez dans le paramètre optionnel *tabVariables* le nom du tableau de pointeurs devant être rempli avec des pointeurs vers les variables ou champs associés aux objets. Si un objet n'a pas de variable associée, le pointeur Nil est retourné. Dans le cas d'un objet de type "sous-formulaire", un pointeur sur la table du sous-formulaire est retourné.

Le troisième tableau (facultatif), *tabPages*, est rempli avec les numéros de pages du formulaire. Chaque ligne de ce tableau contient le numéro de la page sur laquelle se trouve l'objet correspondant.

Les objets provenant d'un formulaire hérité sont considérés comme appartenant à la page 0 du formulaire courant.

Le paramètre optionnel \* permet de réduire la liste des objets retournés à la page courante du formulaire. Lorsque ce paramètre est passé, seuls les objets de la page courante, de la page 0 et des pages héritées sont retournés par la commande. Autrement dit, tous les objets présents dans la page courante du formulaire (visibles ou non) sont traités par la commande.

## Commandes modifiées

Les commandes suivantes étaient auparavant situées dans le thème "Table".

### FORMULAIRE ENTREE

FORMULAIRE ENTREE({table; }formulaire{; *formUtilisateur*}; \*)

Paramètres	Type	Description
table	Table	→ Table pour laquelle définir le formulaire entrée ou Table par défaut si ce paramètre est omis
formulaire	Alpha	→ Nom du formulaire à utiliser
<i>formUtilisateur</i>	Alpha	→ <i>Nom du formulaire utilisateur à utiliser</i>
*	*	→ Taille de fenêtre automatique

La commande FORMULAIRE ENTREE admet le nouveau paramètre facultatif *formUtilisateur*, permettant de désigner un formulaire utilisateur comme formulaire entrée par défaut.

*Note* Pour plus d'informations sur les formulaires utilisateurs, reportez-vous au [paragraphe "Formulaires utilisateurs", page 254](#).

Si vous passez un nom de formulaire utilisateur valide dans le paramètre *formUtilisateur*, ce formulaire sera utilisé par défaut dans le process courant.

Ce principe permet de disposer simultanément de différents formulaires utilisateurs personnalisés (générés à l'aide de la commande [CREER FORMULAIRE UTILISATEUR](#)) et d'utiliser celui qui convient en fonction du contexte.

- ▼ Dans une base de facturation gérant plusieurs sociétés, la création d'une facture doit s'effectuer dans le formulaire utilisateur correspondant :

#### Au cas ou

: (société="4D SA")

**FORMULAIRE ENTREE**([Factures];"Saisie";"4D\_SA")

: (société="4D Inc")

**FORMULAIRE ENTREE**([Factures];"Saisie";"4D\_Inc")

: (société="Acme")

**FORMULAIRE ENTREE**([Factures];"Saisie";"ACME")

#### Fin de cas

**AJOUTER ENREGISTREMENT**([Factures])

*Référence* : [FORMULAIRE SORTIE](#)

## FORMULAIRE SORTIE

FORMULAIRE SORTIE({table; }formulaire{; *formUtilisateur*})

Paramètres	Type	Description
table	Table	→ Table pour laquelle définir le formulaire sortie ou Table par défaut si ce paramètre est omis
formulaire	Alpha	→ Nom du formulaire à utiliser
<i>formUtilisateur</i>	Alpha	→ <i>Nom du formulaire utilisateur à utiliser</i>

La commande FORMULAIRE SORTIE admet le nouveau paramètre facultatif *formUtilisateur*, permettant de désigner un formulaire utilisateur comme formulaire sortie par défaut.

*Note* Pour plus d'informations sur les formulaires utilisateurs, reportez-vous au [paragraphe "Formulaires utilisateurs", page 254](#).

Si vous passez un nom de formulaire utilisateur valide dans le paramètre *formUtilisateur*, ce formulaire sera utilisé par défaut dans le process courant.

Ce principe permet de disposer simultanément de différents formulaires utilisateurs personnalisés (générés à l'aide de la commande [CREER FORMULAIRE UTILISATEUR](#)) et d'utiliser celui qui convient en fonction du contexte.

**Référence :** [FORMULAIRE ENTREE](#)

## Liens

4<sup>e</sup> Dimension 2004 permet un contrôle accru sur les liens par programmation :

- il est désormais possible de lire et de fixer individuellement le statut de chaque lien à l'aide des nouvelles commandes [LIRE LIEN CHAMP](#) et [FIXER LIEN CHAMP](#).
- la nouvelle commande [LIRE LIENS AUTOMATIQUES](#) permet de connaître le statut courant de chaque lien de la base.

*Note* Pour des raisons de cohérence, la commande existante LIENS AUTOMATIQUES a été renommée FIXER LIENS AUTOMATIQUES. Son fonctionnement est inchangé.

## LIRE LIEN CHAMP

LIRE LIEN CHAMP(champN; aller; retour{; \*})

Paramètres	Type	Description
champN	Champ	→ Champ de départ du lien
aller	Entier long	← Statut du lien aller
retour	Entier long	← Statut du lien retour
*	*	→ <ul style="list-style-type: none"> <li>• Si passé : aller et retour retournent le statut courant effectif du lien (valeurs 2 ou 3 uniquement)</li> <li>• Si omis (défaut) : aller et retour peuvent retourner la valeur 1 si le lien n'a pas été modifié par programmation</li> </ul>

La commande LIRE LIEN CHAMP permet de connaître le statut automatique/manuel du lien partant du *champN* pour le process courant.

Tous les liens peuvent être consultés, y compris les liens déclarés automatiques dans la fenêtre de Structure.

- Passez dans *champN* le nom du champ de la table N d'où part le lien dont vous souhaitez connaître le statut.  
Si aucun lien ne part du champ *champN*, les paramètres *aller* et *retour* retournent 0, une erreur est générée et la variable système *OK* prend la valeur 0 (cf. ci-dessous).
- Après l'exécution de la commande, la variable *aller* contient une valeur indiquant si le lien aller spécifié est défini comme automatique :
  - 0 = il n'y a pas de lien partant de *champN*. L'erreur de syntaxe n°16 ("Ce champ ne possède pas de lien") est générée et la variable système *OK* prend la valeur 0.
  - 1 = le statut automatique/manuel du lien aller spécifié est celui défini par l'option **Lien aller auto** dans les propriétés du lien en mode Structure (il n'a pas été modifié par programmation).
  - 2 = le lien N vers 1 est manuel pour le process.
  - 3 = le lien N vers 1 est automatique pour le process.

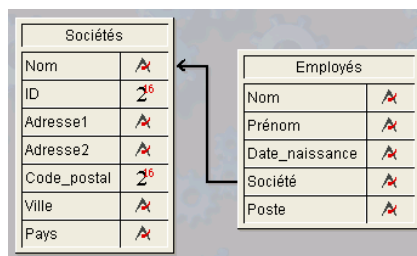


- Après l'exécution de la commande, la variable *retour* contient une valeur indiquant si le lien retour spécifié est défini comme automatique :
  - 0 = il n'y a pas de lien partant de *champN*. L'erreur de syntaxe n°16 ("Ce champ ne possède pas de lien") est générée et la variable système *OK* prend la valeur 0.
  - 1 = le statut automatique/manuel du lien retour spécifié est celui défini par l'option **Lien retour auto** dans les propriétés du lien en mode Structure (il n'a pas été modifié par programmation).
  - 2 = le lien 1 vers N est manuel pour le process.
  - 3 = le lien 1 vers N est automatique pour le process.

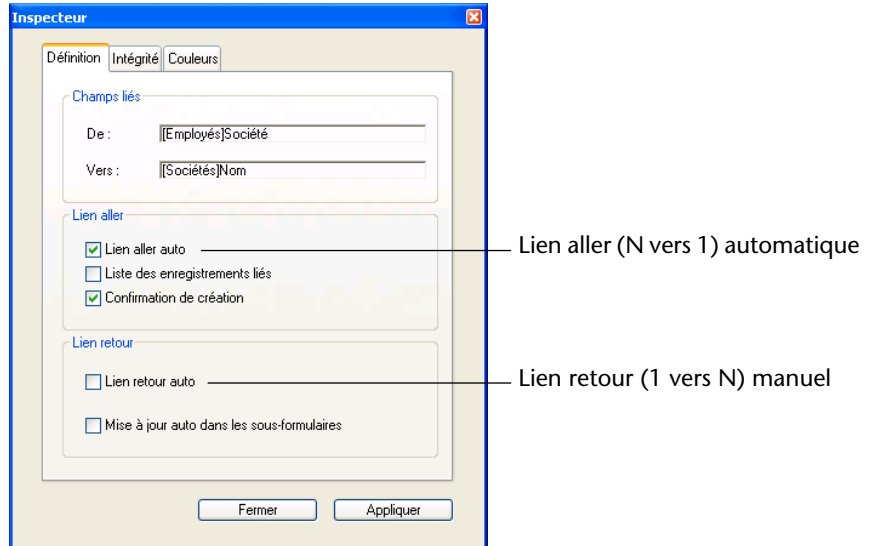
Vous pouvez comparer les valeurs reçues dans les paramètres *aller* et *retour* aux constantes du thème "Liens" :

Constante	Type	Valeur
Pas de lien	Entier long	0
Configuration Structure	Entier long	1
Manuel	Entier long	2
Automatique	Entier long	3

- Le paramètre optionnel \* permet de "forcer" la lecture du statut courant du lien, même s'il n'a pas été modifié par programmation. Autrement dit, lorsque vous passez le paramètre \*, seules les valeurs 2 ou 3 peuvent être retournées dans les paramètres *aller* et *retour*.
- ▼ Soit la structure suivante :



Les propriétés du lien reliant le champ [Employés]Société au champ [Sociétés]Nom sont les suivantes :



Ce code illustre les différentes possibilités offertes par les commandes LIRE LIEN CHAMP et FIXER LIEN CHAMP ainsi que leurs effets :

**C\_BOOLEEN**(liens\_Appel;liens\_Retour)

**C\_ENTIER LONG**(aller,retour)

**LIRE LIENS AUTOMATIQUES**(liens\_Appel;liens\_Retour)

    `retourne Faux, Faux

**LIRE LIEN CHAMP**([Employés]Société;aller;retour) `retourne 1,1

**LIRE LIEN CHAMP**([Employés]Société;aller;retour;\*) `retourne 3,2

**FIXER LIEN CHAMP** ([Employés]Société;2;0)

    `passe le lien N vers 1 en manuel

**LIRE LIEN CHAMP**([Employés]Société;aller;retour) `retourne 2,1

**LIRE LIEN CHAMP**([Employés]Société;aller;retour;\*) `retourne 2, 2

**FIXER LIEN CHAMP** ([Employés]Société;1;0)

    `rétablit les paramètres définis en

    `structure pour le lien N vers 1

**LIRE LIEN CHAMP**([Employés]Société;aller;retour) `retourne 1,1

**LIRE LIEN CHAMP**([Employés]Société;aller;retour;\*) `retourne 3,2

**FIXER LIENS AUTOMATIQUES**(Vrai;Vrai) `passe tous les liens de toutes les

    `tables en automatique

**LIRE LIENS AUTOMATIQUES**(liens\_Appel;liens\_Retour) `retourne Vrai, Vrai  
**LIRE LIEN CHAMP**([Employés]Société;aller;retour) `retourne 1,1  
**LIRE LIEN CHAMP**([Employés]Société;aller;retour;\*) `retourne 3,3

- ▼ Voici un exemple d'utilisation type des deux commandes LIRE LIEN CHAMP et FIXER LIEN CHAMP :

**C\_ENTIER LONG**(aller;retour)  
`lecture des paramètres courants  
**LIRE LIEN CHAMP**([MaTable]MonChamp;aller;retour)  
`passage en auto du lien aller (N vers 1) et conservation en manuel du lien  
`retour (1 vers N)  
**FIXER LIEN CHAMP**([MaTable]MonChamp;3;2)  
**CHERCHER** (...) `réalisation d'une recherche  
`rétablissement des paramètres originels  
**FIXER LIEN CHAMP**([MaTable]MonChamp;aller;retour)

## FIXER LIEN CHAMP

**FIXER LIEN CHAMP**(tableN | champN; aller; retour)

Paramètres	Type	Description
tableN	Table	→ Table de départ des liens ou
champN	Champ	Champ de départ du lien
aller	Entier long	→ Statut du lien aller partant du champ ou des liens aller partant de la table
retour	Entier long	→ Statut du lien retour partant du champ ou des liens retour partant de la table

La commande **FIXER LIEN CHAMP** permet de définir séparément le statut automatique/manuel de chaque lien de la base, quel que soit son statut initial défini en mode Structure dans la fenêtre de paramétrage des liens.

- Passez dans le premier paramètre un nom de table ou de champ :
  - si vous passez un nom de champ (*champN*), la commande s'appliquera uniquement au lien partant du champ N désigné.
  - si vous passez un nom de table (*tableN*), la commande s'appliquera à tous les liens partant de la table N désignée.
  - si aucun lien ne part du champ *champN* ou de la table *tableN*, les paramètres *aller* et *retour* retournent 0, l'erreur de syntaxe n°16 ("Ce champ ne possède pas de lien") est générée et la variable système *OK* prend la valeur 0.

- Passez dans le paramètre *aller* une valeur indiquant la modification du statut automatique/manuel à appliquer au(x) lien(s) de type N vers 1 — c'est-à-dire, au(x) lien(s) aller — désigné(s). Vous pouvez utiliser les constantes du thème "Liens" :
  - Ne pas changer (0) = ne pas modifier le statut courant du ou des lien(s) aller.
  - Configuration Structure (1) = utiliser le paramétrage défini pour le(s) lien(s) aller dans la fenêtre de Structure de l'application.
  - Manuel (2) = rendre manuel(s) le(s) lien(s) aller pour le process courant.
  - Automatique (3) = rendre automatique(s) le(s) lien(s) aller pour le process courant.
- Passez dans le paramètre *retour* une valeur indiquant la modification du statut automatique/manuel à appliquer au(x) lien(s) de type 1 vers N — c'est-à-dire, au(x) lien(s) retour — désigné(s). Vous pouvez utiliser les constantes du thème "Liens" :
  - Ne pas changer (0) = ne pas modifier le statut courant du ou des lien(s) retour.
  - Configuration Structure (1) = utiliser le paramétrage défini pour le(s) lien(s) retour dans la fenêtre de Structure de l'application.
  - Manuel (2) = rendre manuel(s) le(s) lien(s) retour pour le process courant.
  - Automatique (3) = rendre automatique(s) le(s) lien(s) retour pour le process courant.

---

*Note* Les modifications effectuées à l'aide de cette commande s'appliquent au process courant uniquement. Le paramétrage des liens défini à l'aide des options de la fenêtre des propriétés n'est pas modifié.

---

- ▼ Cette nouvelle commande simplifie la gestion des liens avec l'éditeur d'états rapides. Dans les versions précédentes de 4<sup>e</sup> Dimension, pour utiliser les liens automatiques autres que ceux définis en mode Structure, il était nécessaire de passer tous les liens en automatique. Désormais, le code suivant permet de n'utiliser que les liens définis :

**FIXER LIENS AUTOMATIQUES(Faux;Faux) `Initialisation des liens`  
`Seuls les liens suivants seront utilisés`**

**FIXER LIEN CHAMP([Facture]ID\_Client;Automatique;Automatique)**

**FIXER LIEN CHAMP([Ligne\_Facture]ID\_Facture;Automatique;**

**Automatique)**

**QR ETAT([Facture];Caractere(1);Vrai;Vrai;Vrai)**

## LIRE LIENS AUTOMATIQUES

LIRE LIENS AUTOMATIQUES(aller; retour)

Paramètres	Type	Description
aller	Booléen	← Statut des liens de N vers 1
retour	Booléen	← Statut des liens de 1 vers N

La commande LIRE LIENS AUTOMATIQUES permet de savoir si le statut automatique/manuel de tous les liens manuels N vers 1 et 1 vers N de la base a été modifié dans le process courant. Cette commande représente l'inverse de la commande LIENS AUTOMATIQUES (rebaptisée FIXER LIEN AUTOMATIQUE dans 4<sup>e</sup> Dimension 2004).

- *aller* : ce paramètre retourne Vrai si un appel antérieur de la commande FIXER LIENS AUTOMATIQUES a rendu automatiques tous les liens manuels N vers 1 — par exemple FIXER LIENS AUTOMATIQUES(Vrai;Faux).  
Ce paramètre retourne Faux si la commande FIXER LIENS AUTOMATIQUES n'a pas été appelée ou si sa précédente exécution n'a pas modifié les liens manuels N vers 1 — par exemple FIXER LIENS AUTOMATIQUES(Faux;Faux).
- *retour* : ce paramètre retourne Vrai si l'appel précédent de la commande FIXER LIENS AUTOMATIQUES a rendu automatiques tous les liens manuels 1 vers N — par exemple FIXER LIENS AUTOMATIQUES(Vrai;Vrai).  
Ce paramètre retourne Faux si la commande FIXER LIENS AUTOMATIQUES n'a pas été appelée ou si sa précédente exécution n'a pas modifié les liens manuels 1 vers N — par exemple FIXER LIENS AUTOMATIQUES(Vrai;Faux).
- ▼ Reportez-vous au premier exemple de la commande [LIRE LIEN CHAMP](#).

## Événements formulaires

4<sup>e</sup> Dimension 2004 propose plusieurs nouveaux événements formulaires. Ces nouveaux événements répondent aux différents types de besoins suivants :

- la gestion du survol des objets,
- la gestion des boutons 3D,
- la gestion du nouvel objet "List box",
- la gestion des enregistrements en liste,
- la gestion des listes hiérarchiques.

## Événements formulaire liés aux survols

Trois nouveaux événements formulaires permettent de prendre en charge le survol (*roll over*) par le curseur de la souris de tout objet de formulaire :

- **Sur début survol**
- **Sur survol**
- **Sur fin survol**

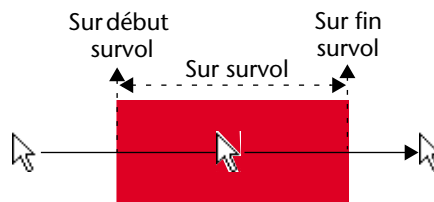
Dans les versions précédentes de 4<sup>e</sup> Dimension, la gestion du survol nécessitait l'écriture de méthodes complexes. Dans 4<sup>e</sup> Dimension 2004, l'emploi des événements **Sur début survol**, **Sur survol** et **Sur fin survol** simplifie cette gestion.

Ces événements sont générés pour les fenêtres actives de l'ensemble des process ayant une interface. En cas de superposition d'objets, l'événement est généré par le premier objet capable de le gérer dans l'ordre des plans du haut vers le bas.

Les objets rendus invisibles par la commande CHOIX VISIBLE ne génèrent pas ces événements. Pendant la saisie d'un objet, les autres objets peuvent recevoir les événements de survol en fonction de la position de la souris.

Si ces événements sont cochés pour le formulaire, ils sont générés pour chaque objet du formulaire. S'ils sont cochés pour un objet, ils sont générés pour cet objet uniquement.

- L'événement Sur début survol se produit lorsque la souris rentre dans la zone graphique d'un objet. La méthode de l'objet est appelée une fois.
- L'événement Sur survol se produit à chaque fois que la souris bouge (d'au moins un pixel) alors qu'elle se trouve dans la zone graphique d'un objet. La méthode de l'objet est appelée à chaque fois.
- L'événement Sur fin survol se produit lorsque la souris sort de la zone graphique d'un objet. La méthode de l'objet est appelée une fois.



## Événements formulaire liés aux boutons 3D

4<sup>e</sup> Dimension 2004 introduit une nouvelle famille de boutons 3D autorisant la mise en place d'interfaces graphiques avancées (pour une description de ces nouveaux boutons 3D, reportez-vous au [paragraphe "Modification des boutons 3D", page 155](#)).

Deux nouveaux événements formulaires ont été créés pour la gestion de ces nouveaux objets : **Sur clic long** et **Sur clic flèche**.

- **Sur clic long** : cet événement est généré lorsqu'un bouton 3D reçoit un clic et que le bouton de la souris reste enfoncé pendant un certain laps de temps. En pratique, le délai à l'issue duquel l'événement est généré est égal au délai maximal séparant un double-clic, tel qu'il a été défini dans les préférences du système.

Cet événement peut être généré pour tous les styles de boutons 3D, boutons radio 3D et cases à cocher 3D, à l'exception des boutons 3D "ancienne génération" (style *Décalage du fond*) et des zones de flèche des boutons 3D avec pop up menu (cf. ci-dessous). Il doit être coché dans les Propriétés des objets pour pouvoir être généré.

Cet événement est généralement utilisé pour afficher des pop up menus en cas de clics longs sur des boutons. L'événement **Sur clic**, s'il est activé, est généré si l'utilisateur relâche le bouton de la souris avant le délai du "clic long".

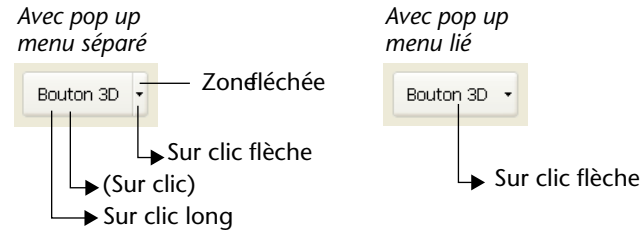
- **Sur clic flèche** : certains styles de boutons 3D peuvent être associés à un pop up menu et afficher une flèche. Un clic sur cette flèche fait généralement apparaître un pop up menu de sélection proposant des actions supplémentaires en rapport avec l'action principale du bouton.

4<sup>e</sup> Dimension vous permet de gérer ce type de bouton à l'aide du nouvel événement **Sur clic flèche**. Cet événement est généré lorsque l'utilisateur clique sur la "flèche" (dès que le bouton de la souris est enfoncé) :

- si le pop up menu est de type "Séparé", l'événement est généré uniquement en cas de clic sur la zone fléchée du bouton.

- si le pop up menu est de type “Lié”, l’événement est généré en cas de clic sur n’importe quelle partie du bouton. A noter qu’avec ce type de bouton, l’événement **Sur clic long** ne peut pas être généré.

Exemple de bouton 3D (style “Bouton barre outils” Windows)



Les styles de boutons 3D, boutons radio 3D et cases à cocher 3D acceptant la propriété “Avec pop up menu” sont les suivants :

- Aucun
- Bouton barre outils
- Bevel
- Bevel arrondi
- Office XP

## Événements formulaire liés aux list box

Les list box sont un nouveau type d’objet proposé dans 4<sup>e</sup> Dimension 2004. Pour plus d’informations sur la création et le paramétrage des objets de type List box dans l’éditeur de formulaires, reportez-vous au [paragraphe “List Box”, page 133](#). Pour plus d’informations sur leur gestion programmée, reportez-vous au [paragraphe “List box”, page 328](#).

Sept nouveaux événements formulaires permettent de prendre en charge les spécificités des list box : **Sur avant saisie**, **Sur nouvelle sélection**, **Sur déplacement colonne**, **Sur déplacement ligne**, **Sur redimensionnement colonne**, **Sur clic entête** et **Sur après tri**.

Ces événements sont disponibles pour l’objet List box. Ils permettent de gérer diverses situations.

- **Sur avant saisie** : cet événement est généré juste avant qu’une cellule de list box passe en mode édition (c’est-à-dire, avant que le curseur de saisie soit affiché). Cet événement permet au développeur, par exemple, d’afficher un texte différent selon que l’utilisateur est en mode affichage ou édition.



- **Sur nouvelle sélection** : cet événement est généré à chaque fois que la sélection courante de lignes ou de colonnes de la list box est modifiée.

*Note* Cet événement est également généré dans les listes d'enregistrements (cf. [paragraphe "Événements formulaire liés aux enregistrements en liste", page 306](#)) et les listes hiérarchiques (cf. [paragraphe "Événements formulaire liés aux listes hiérarchiques", page 307](#)).

- **Sur déplacement colonne** : cet événement est généré lorsqu'une colonne de list box est déplacée par l'utilisateur via le glisser-déposer. Il n'est pas généré si la colonne est glissée et déposée à son emplacement initial.  
La commande `NUMERO COLONNE LISTBOX DEPLACEE` permet de connaître le nouvel emplacement de la colonne.
- **Sur déplacement ligne** : cet événement est généré lorsqu'une ligne de list box est déplacée par l'utilisateur via le glisser-déposer. Il n'est pas généré si la ligne est glissée et déposée à son emplacement initial.  
La commande `NUMERO LIGNE LISTBOX DEPLACEE` permet de connaître le nouvel emplacement de la colonne.
- **Sur redimensionnement colonne** : cet événement est généré lorsque la largeur d'une colonne de list box est modifiée (via la souris ou par programmation à l'aide de la commande `FIXER LARGEUR COLONNE LISTBOX`).
- **Sur clic entête** : cet événement est généré lorsqu'un clic se produit sur l'en-tête d'une colonne de list box.

*Note* Dans ce cas, la commande `Self` permet de connaître l'en-tête de colonne sur laquelle le clic s'est produit (cf. [paragraphe "Self", page 332](#)).

Si la propriété **Triable** a été cochée pour la list box, il est possible d'autoriser ou non le tri standard sur la colonne en passant la valeur 0 ou -1 dans la variable `$0` :

- Si `$0` vaut 0, le tri standard est effectué.
- Si `$0` vaut -1, le tri standard n'est pas effectué et l'en-tête n'affiche pas la flèche de tri. Le développeur peut toutefois générer un tri des colonnes sur des critères personnalisés à l'aide des commandes de gestion des tableaux de 4<sup>e</sup> Dimension.

Si la propriété **Triable** n'a pas été cochée pour la list box, la variable `$0` n'est pas utilisée.

- **Sur après tri** : cet événement est généré juste après qu'un tri standard ait été effectué (par conséquent, il n'est pas généré si \$0 retourne -1 dans l'événement **Sur clic entête**). Ce mécanisme est utile pour conserver le sens du dernier tri effectué par l'utilisateur. Dans cet événement, la commande Self retourne un pointeur sur la variable de la colonne ayant été triée.

---

*Note* Dans le cadre de l'utilisation d'une list box, l'événement **Sur clic** est généré lorsqu'un clic droit (ou Control+clic sous MacOS) se produit sur une colonne ou un en-tête de colonne.

---

## Événements formulaire liés aux enregistrements en liste

La gestion des enregistrements affichés sous forme de liste (en mode Utilisation, à l'aide des commandes MODIFIER SELECTION / VISUALISER SELECTION ou dans les sous-formulaires) a été modifiée dans 4<sup>e</sup> Dimension 2004. Toutes les nouveautés sont détaillées dans le [paragraphe "Gestion des listes d'enregistrements"](#), page 165.

Deux nouveaux événements formulaires permettent de mieux prendre en charge l'affichage de ces enregistrements : **Sur nouvelle sélection** et **Sur chargement ligne**.

- **Sur nouvelle sélection** : cet événement est généré à chaque fois que l'enregistrement courant ou la sélection courante de lignes est modifié(e) dans un formulaire en liste ou un sous-formulaire.

---

*Note* Cet événement est également généré dans les objets list box (cf. [paragraphe "Événements formulaire liés aux list box"](#), page 304) et les listes hiérarchiques (cf. [paragraphe "Événements formulaire liés aux listes hiérarchiques"](#), page 307).

---

- **Sur chargement ligne** : cet événement est généré en mode Saisie en liste, lorsqu'un enregistrement est chargé en modification (l'utilisateur a cliqué sur la ligne de l'enregistrement et un champ passe en édition).

## Sur affichage corps / Sur entête

L'appel depuis les événements Sur affichage corps et Sur entête d'une commande 4D provoquant l'affichage d'une boîte de dialogue est désormais interdit et provoque une erreur de syntaxe.

Les commandes concernées sont notamment : ALERTE, DIALOGUE, CONFIRMER, Demander, AJOUTER ENREGISTREMENT, MODIFIER ENREGISTREMENT, VISUALISER SELECTION et MODIFIER SELECTION.

## Événements formulaire liés aux listes hiérarchiques

La gestion des listes hiérarchiques a été facilitée dans 4<sup>e</sup> Dimension 2004 via la modification de nombreuses commandes (cf. [paragraphe “Listes hiérarchiques”, page 260](#)).

En outre, trois nouveaux événements formulaires permettent de mieux prendre en charge les actions utilisateurs effectuées sur les listes hiérarchiques : **Sur nouvelle sélection**, **Sur déployer** et **Sur contracter**.

- **Sur nouvelle sélection** : cet événement est généré à chaque fois que la sélection dans la liste hiérarchique est modifiée à la suite d’un clic ou de la frappe d’une touche au clavier.

*Note* Cet événement est également généré dans les objets list box (cf. [paragraphe “Événements formulaire liés aux list box”, page 304](#)) et les listes d’enregistrements (cf. [paragraphe “Événements formulaire liés aux enregistrements en liste”, page 306](#)).

- **Sur déployer** : cet événement est généré à chaque fois qu’un élément de la liste hiérarchique a été déployé via un clic ou une touche du clavier.
- **Sur contracter** : cet événement est généré à chaque fois qu’un élément de la liste hiérarchique a été contracté via un clic ou une touche du clavier.

Ces événements ne sont pas mutuellement exclusifs. Ils peuvent être générés les uns après les autres pour une liste hiérarchique.

### *Suite à la frappe d’une touche clavier (dans l’ordre) :*

Événement	Contexte
Sur données modifiées	Un élément était en édition
Sur déployer / Sur contracter	Ouverture/fermeture de sous-liste à l’aide des touches fléchées → ou ←
Sur nouvelle sélection	Sélection d’un nouvel élément
Sur clic	Activation de la liste par le clavier

### *Suite à un clic souris (dans l’ordre) :*

Événement	Contexte
Sur données modifiées	Un élément était en édition

Sur déployer / Sur contracter	Ouverture/fermeture de sous-liste via un clic sur l'icône de déploiement/contraction <i>ou bien</i> Double-clic sur une sous-liste non éditable
Sur nouvelle sélection	Sélection d'un nouvel élément
Sur clic / Sur double clic	Activation de la liste par un clic ou un double-clic

## Interface utilisateur

Ce chapitre décrit les nouvelles commandes [LIRE TITRES TABLES](#), [LIRE TITRES CHAMPS](#) et [DEFILER LIGNES](#) ainsi que les modifications apportées aux commandes [FIXER TITRES CHAMPS / FIXER TITRES TABLES](#), [Pop up menu](#) et [Objet focus](#) (anciennement Dernier objet).

### **LIRE TITRES TABLES** LIRE TITRES TABLES(*titresTables*; *numTables*)

Paramètres	Type	Description
<i>titresTables</i>	Tableau Texte	← Noms courants des tables
<i>numTables</i>	Tableau Entier long	← Numéros des tables

La commande LIRE TITRES TABLES remplit les tableaux *titresTables* et *numTables* avec les noms et les numéros des tables de la base définis dans la fenêtre de Structure ou via la commande FIXER TITRES TABLES. Le contenu des deux tableaux est synchronisé.

Si la commande FIXER TITRES TABLES a été appelée lors de la session, LIRE TITRES TABLES retourne uniquement les noms "modifiés" et les numéros des tables ayant été définies via cette commande. Sinon, LIRE TITRES TABLES retourne le nom défini en mode Structure de toutes les tables de la base.

Dans les deux cas, la commande ne retourne pas les tables déclarées invisibles.

**Référence :** [LIRE TITRES CHAMPS](#), [FIXER TITRES TABLES](#).

## LIRE TITRES CHAMPS

LIRE TITRES CHAMPS(table; titresChamps; numChamps)

Paramètres	Type	Description
table	Table	→ Table dont vous souhaitez connaître les noms des champs
titresChamps	Tableau Texte	← Noms courants des champs
numChamps	Tableau Entier long	← Numéros des champs

La commande LIRE TITRES CHAMPS remplit les tableaux *titresChamps* et *numChamps* avec les noms et les numéros des champs de la *table* désignée. Le contenu des deux tableaux est synchronisé.

Si la commande FIXER TITRES CHAMPS a été appelée au cours de la session, LIRE TITRES CHAMPS retourne uniquement les noms “modifiés” et les numéros des champs ayant été définis via cette commande. Sinon, LIRE TITRES CHAMPS retourne le nom défini dans la fenêtre de Structure de tous les champs de la base. Dans les deux cas, la commande ne retourne pas les champs déclarés invisibles.

**Référence :** LIRE TITRES TABLES, FIXER TITRES CHAMPS

## DEFILER LIGNES

DEFILER LIGNES({\*; }objet{; position}{;\* })

Paramètres	Type	Description
*		→ Si spécifié, objet est un nom d’objet (chaîne) Si omis, objet est une table
objet	Objet de formulaire	→ Nom d’objet (si * est spécifié) ou Table (si * est omis)
position	Entier long	→ Position d’enregistrement dans la sélection
*		→ Afficher la ligne en première position après défilement

La commande DEFILER LIGNES permet de faire défiler les lignes d’un formulaire liste (affiché via la commande MODIFIER SELECTION ou VISUALISER SELECTION) ou d’un sous-formulaire de manière à afficher le premier enregistrement sélectionné ou un enregistrement spécifique.

Si vous passez le premier paramètre optionnel \*, vous indiquez que le paramètre *objet* est le nom d’un objet sous-formulaire (dans ce cas, passez une chaîne dans *objet*).

Si vous ne passez pas ce paramètre, vous indiquez que le paramètre *objet* est une table (table du formulaire liste ou du sous-formulaire).

Si vous ne passez pas le paramètre *position*, la commande provoque le défilement vertical des lignes de la liste de manière à ce que le premier enregistrement sélectionné (surligné) dans la liste soit visible.

Si aucun enregistrement n'est sélectionné, la commande ne fait rien. Si au moins un enregistrement sélectionné est visible, la commande ne fait rien.

Le paramètre *position* vous permet d'indiquer le numéro d'un enregistrement parmi la sélection courante, c'est-à-dire sa position ; si vous passez ce paramètre, la commande provoque le défilement vertical des lignes de la liste de manière à ce que l'enregistrement désigné soit visible (qu'il soit surligné ou non). Si l'enregistrement est déjà visible, la commande ne fait rien.

Si vous passez le second paramètre optionnel \*, la ligne rendue visible par la commande (si la liste a effectivement défilé) sera placée en première position de la liste.

---

*Note* La commande **MARQUER ENREGISTREMENTS** comporte désormais un paramètre \* facultatif permettant de déléguer la gestion du défilement à la commande **DEFILER LIGNES**.

---

**Référence : MARQUER ENREGISTREMENTS**

### **FIXER TITRES CHAMPS / FIXER TITRES TABLES**

FIXER TITRES CHAMPS (table | sous-table; titresChamps; numChamps)  
FIXER TITRES TABLES (titresTables; numTables)

Dans 4<sup>e</sup> Dimension 2004, les noms de champs et de tables définis par les commandes **FIXER TITRES CHAMPS** et **FIXER TITRES TABLES** sont affichés dans le nouvel éditeur de formules (cf. [paragraphe "Editeur de formules", page 35](#)). Par conséquent, désormais ces noms ne peuvent plus contenir de caractères "interdits" dans les méthodes (tels que -?\*, etc.)

## Pop up menu

Pop up menu (contenu{; parDéfaut{; coordX; coordY}) → Numérique

Paramètres	Type	Description
contenu	Texte	→ Définition du texte du menu
parDéfaut	Numérique	→ Numéro de l'élément sélectionné par défaut
coordX	Numérique	→ Coordonnée X du coin supérieur gauche
coordY	Numérique	→ Coordonnée Y du coin supérieur gauche
Résultat	Numérique	← Numéro de l'élément de menu sélectionné

La commande Pop up menu admet deux nouveaux paramètres facultatifs, permettant de désigner avec précision l'emplacement du coin supérieur gauche du pop up menu à afficher.

Passez respectivement dans *coordX* et *coordY* les coordonnées horizontale et verticale de l'angle supérieur gauche du menu. Ces coordonnées doivent être exprimées en pixels dans le système de coordonnées local au formulaire courant. Ces deux paramètres doivent être passés ensemble ; si un seul est passé, il est ignoré.

Si vous utilisez les paramètres *coordX* et *coordY*, le paramètre *parDéfaut* est ignoré. Dans ce cas en effet, la souris ne se trouve pas nécessairement au niveau du pop up menu.

Ces nouveaux paramètres sont utiles notamment pour la gestion des boutons 3D avec pop up menu associé (cf. [paragraphe "Modification des boutons 3D", page 155](#)).

## Objet focus

**Note préliminaire :** Objet focus est le nouveau nom de la commande existante Dernier objet. Cette commande a été renommée par souci de clarté.

Objet focus → Pointeur

La commande Objet focus retourne un pointeur vers l'objet ayant le focus dans le formulaire courant. Si aucun objet n'a le focus, la commande retourne Nil.

Lorsqu'elle est utilisée avec un objet de type List box, la fonction Objet focus retourne un pointeur vers la colonne (le tableau) de la list box ayant le focus. Pour plus d'informations, reportez-vous au [paragraphe "Objet focus", page 332](#).

## Gestion de la saisie

Ce thème comporte la nouvelle commande **EDITER ELEMENT**, permettant de passer en mode édition tout élément d'un objet de type liste (liste hiérarchique, list box et sous-formulaire).

### EDITER ELEMENT

EDITER ELEMENT({\*; } objet{; élément})

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

Cette commande permet de passer en "mode édition" l'élément courant ou l'élément de numéro *élément* du tableau ou de la liste désigné(e) par le paramètre *objet*.

Le mode édition signifie que l'élément est sélectionné et prêt à être modifié : la saisie d'un caractère remplacera intégralement le contenu de l'élément.

Si vous passez le paramètre optionnel \*, vous indiquez que le paramètre *objet* est un nom d'objet (dans ce cas, passez une chaîne dans *objet*). Si vous ne passez pas le paramètre, vous indiquez que le paramètre *objet* est un champ ou une variable. Dans ce cas, vous ne passez pas une chaîne mais une référence de champ ou de variable.

Cette commande s'applique aux objets saisissables suivants :

- Listes hiérarchiques,
- List box (cf. [paragraphe "List box", page 328](#)),
- Sous-formulaires (dans ce cas, seul un nom d'objet — le sous-formulaire — peut être passé dans *objet*),
- Formulaire liste affichés via la commande MODIFIER SELECTION ou VISUALISER SELECTION.



Si la commande est utilisée avec un objet saisissable qui n'est pas une liste, elle équivaut à la commande ALLER A CHAMP.

La commande ne fait rien si la liste ou le tableau désigné(e) est vide ou invisible. Si la liste ou le tableau n'est pas saisissable, la commande sélectionne (sans passer en édition) l'élément spécifié.

Le paramètre facultatif *élément* vous permet de désigner la position de l'élément (liste hiérarchique) ou le numéro de la ligne (list box, formulaire liste et sous-formulaire en mode "multi-sélection") à passer en édition. Si vous ne passez pas ce paramètre, la commande s'applique à l'élément courant de *objet*. S'il n'y a pas d'élément courant, le premier élément de *objet* passe en édition.

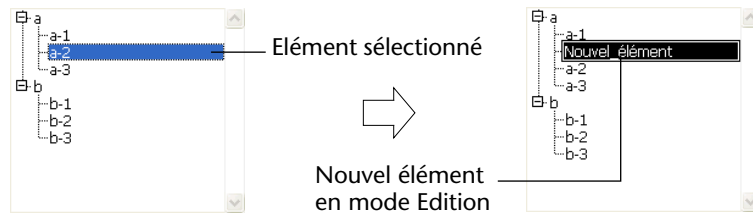
- 
- Notes*
- Dans le cadre des listes hiérarchiques, la commande EDITER ELEMENT provoque automatiquement un redessinment de la liste. Par conséquent, vous ne devez pas appeler la commande REDESSINER LISTE lorsque la commande EDITER ELEMENT est utilisée.
  - Dans le cadre des sous-formulaires et des formulaires liste, la commande passe en édition le premier champ de la ligne spécifiée, dans l'ordre de saisie.
  - Dans le cadre des sous-formulaires en mode "sélection unique", la commande passe en édition la ligne courante, quelle que soit la valeur du paramètre *élément*.
- 

- ▼ Cette commande peut être utile notamment lors de la création d'un nouvel élément de liste hiérarchique. Au moment de l'appel de la commande, le dernier élément ajouté ou inséré dans la liste devient automatiquement éditable, sans que l'utilisateur n'ait à effectuer d'action spécifique.

Le code suivant pourrait être la méthode d'un bouton permettant d'insérer un nouvel élément dans une liste existante. Le libellé "Nouvel\_élément" proposé par défaut est automatiquement placé en mode édition :

```
vlUniqueRef:=vlUniqueRef+1
INSERER ELEMENT(hList;*;"Nouvel_élément";vlUniqueRef)
```

**EDITER ELEMENT(\*,"MaListe")**

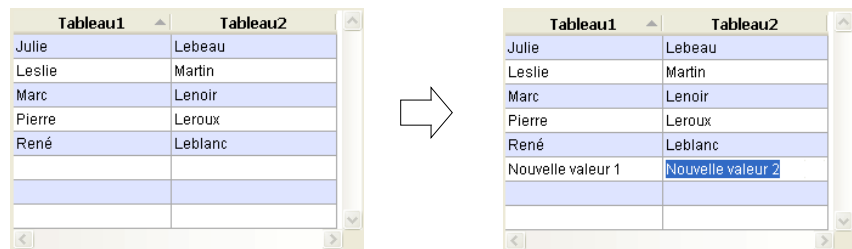


- ▼ Soient deux colonnes d'une list box dont les noms de variables associées sont respectivement "Tableau1" et "Tableau2". L'exemple suivant ajoute une ligne dans la list box et passe le nouvel élément du tableau 2 en mode édition :

```

$vlNumLigne:=Taille tableau(Tableau1)+1
INSERER LIGNE LISTBOX(*,"MaListBox";$vlNumLigne)
Tableau1{$vlNumLigne}:="Nouvelle valeur 1"
Tableau2{$vlNumLigne}:="Nouvelle valeur 2"
    
```

**EDITER ELEMENT(Tableau2; \$vlNumLigne)**



- ▼ L'exemple suivant permet de passer en mode édition le premier champ du dernier sous-enregistrement de la sous-sélection :

```

ALLER A DERNIER SOUS ENREGISTREMENT([Enfants])
EDITER ELEMENT(*,"Sousform")
    
```

## Impressions

### Imprimer sur le serveur

La plupart des commandes d'impression de 4<sup>e</sup> Dimension peuvent désormais être exécutées sur 4D Server sous forme de procédures stockées. Cette nouveauté est utile par exemple pour imprimer des documents en PDF sur le poste serveur.

Les commandes d'impression pouvant être utilisées sont les suivantes :

- IMPRIMER SELECTION
- IMPRIMER ENREGISTREMENT
- IMPRIMER ETIQUETTES
- QR ETAT
- Imprimer ligne

Les commandes de paramétrage d'impression (telles que FIXER OPTION IMPRESSION) sont également utilisables.

Les principes d'utilisation des commandes d'impression sont les suivants :

- Veillez à ce qu'aucune boîte de dialogue n'apparaisse sur le poste serveur (sauf besoin spécifique). Pour cela, il est nécessaire d'appeler les commandes IMPRIMER SELECTION, IMPRIMER ENREGISTREMENT, IMPRIMER ETIQUETTES et QR ETAT avec le paramètre \* (supprimant les boîtes de dialogue d'impression).
- La syntaxe faisant apparaître un éditeur (éditeur d'étiquettes et éditeur d'états rapides) ne fonctionne pas avec 4D Server. Dans ce contexte, la variable système OK prend la valeur 0.
- En cas de problème sur l'imprimante (plus de papier, imprimante déconnectée, etc.), 4<sup>e</sup> Dimension ne génère pas de message d'erreur.

**FIXER OPTION  
IMPRESSION /  
LIRE OPTION  
IMPRESSION  
(thème  
"Impressions")**

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

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

*Note* : Cette nouveauté a été introduite dans 4<sup>e</sup> Dimension 2003.3.

Sous MacOS X, les impressions sont effectuées par défaut en mode PDF. Or, le pilote d'impression PDF ne prend pas en charge les images PICT encapsulant des informations Postscript — ces images sont générées notamment par des logiciels de dessin vectoriel.

Pour résoudre ce problème, une nouvelle *option* a été ajoutée dans 4<sup>e</sup> Dimension, permettant de définir le mode d'impression sous MacOS X pour la session courante. Par défaut, les impressions sont effectuées en mode PDF (mode standard) et le développeur a la possibilité de modifier ce paramétrage en cas de besoin à l'aide des commandes FIXER OPTION IMPRESSION et LIRE OPTION IMPRESSION.

- Option format impression Mac (13) : passez 0 pour fixer l'impression en mode PDF (valeur par défaut) et 1 pour "forcer" l'impression en mode Postscript (impression des PICTs avec postscript encapsulé).

Cette option modifie également le mode d'impression des plug-ins 4D Write, 4D View et 4D Draw.

En revanche, elle n'a pas d'effet sous Windows.

**Important** : Dans les versions de 4<sup>e</sup> Dimension antérieures à la 2003.3, les impressions étaient "forcées" en mode Postscript. Désormais, le mode 0 est utilisé. Si vous souhaitez conserver le mode précédent (notamment si vos bases contiennent des PICT encapsulant du Postscript), il est impératif d'appeler cette option avec la valeur 1.

## Messages

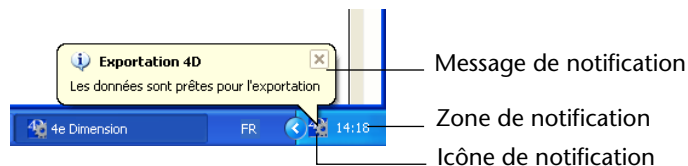
### AFFICHER NOTIFICATION

AFFICHER NOTIFICATION(titre; contenu{; délai})

Paramètres	Type	Description
titre	Alpha 255 →	Titre de la notification
contenu	Alpha 255 →	Texte de la notification
délai	Numérique →	Délai d'affichage en secondes

*Note* Cette commande fonctionne sous Windows uniquement.

La commande AFFICHER NOTIFICATION provoque l'affichage d'un message dans la zone de notification de la barre des tâches de Windows :



Ce type de message est généralement utilisé par le système ou les applications pour informer l'utilisateur d'un événement (déconnexion réseau, disponibilité de mises à jour, etc.)

Passez dans les paramètres *titre* et *contenu* le titre et le texte du message à afficher (dans l'exemple ci-dessus, le titre est "Exportation 4D"). Vous pouvez saisir jusqu'à 255 caractères.

Par défaut, la fenêtre du message reste affichée jusqu'à ce que l'utilisateur clique sur sa case de fermeture. Si vous passez le paramètre facultatif *délai*, la fenêtre sera automatiquement refermée à l'issue de la durée définie si l'utilisateur n'a pas cliqué sur la case de fermeture. A noter que l'icône de notification restera affichée jusqu'à la fin du *délai*, même si l'utilisateur a refermé la fenêtre.

## Saisie

### DIALOGUE

DIALOGUE ({table; }formulaire)

4<sup>e</sup> Dimension 2004 n'affiche plus par défaut les boutons **Annuler** et **Valider** lorsqu'un formulaire utilisé par la commande DIALOGUE est dépourvu de boutons. Dans ce cas, seule la touche **Echap** (Windows) ou **Esc** (MacOS) permet de quitter le formulaire.

*Note* Désormais, la saisie de valeurs dans les champs des formulaires affichés par cette commande est autorisée. Une nouvelle option des Préférences permet de régler ce fonctionnement (cf. [paragraphe "Nouvelles options de compatibilité"](#), page 58).

## Documents système

### Selectionner document

Selectionner document (répertoire; typesFichiers; titre; options{; sélectionnés}}) → Chaîne

Paramètres	Type	Description
répertoire	Texte   Entier long	→ • Chemin d'accès du répertoire à afficher par défaut dans la boîte de dialogue de sélection de document, ou • Chaîne vide pour afficher le dossier d'utilisateur par défaut ("Mes documents" sous Windows, "Documents" sous MacOS), ou • Numéro de chemin d'accès mémorisé
typesFichiers	Texte	→ Liste des types de documents à filtrer, ou * pour ne pas filtrer les documents
titre	Texte	→ Titre de la boîte de dialogue de sélection
options	Entier long	→ Option(s) de sélection, ou 0 pour n'utiliser aucune option

Paramètres	Type	Description
sélectionnés	Tab Texte ←	Tableau contenant la liste des chemins d'accès + noms des fichiers sélectionnés
Résultat	Chaîne ←	Nom du fichier sélectionné (premier fichier de la liste en cas de sélection multiple)

La commande *Selectionner document* affiche une boîte de dialogue standard d'ouverture de documents permettant à l'utilisateur de désigner un ou plusieurs fichier(s) et retourne le nom et/ou le chemin d'accès complet du ou des fichier(s) sélectionné(s).

Le paramètre *répertoire* indique le dossier dont le contenu doit être affiché initialement dans la boîte de dialogue d'ouverture de documents. Vous pouvez passer trois types de valeurs :

- un texte contenant le chemin d'accès complet du dossier à afficher.
- une chaîne vide ("") pour afficher le dossier d'utilisateur par défaut du système ("Mes documents" sous Windows, "Documents" sous MacOS).
- un numéro de chemin d'accès mémorisé (de 1 à 32000) pour afficher le dossier associé.

Avec ce principe, vous pouvez conserver en mémoire le chemin d'accès du dossier ouvert au moment où l'utilisateur a cliqué sur le bouton de sélection, c'est-à-dire le dossier choisi par l'utilisateur. Lors du premier appel d'un numéro arbitraire (par exemple 5), la commande affiche le dossier d'utilisateur par défaut du système (équivalent à passer une chaîne vide). L'utilisateur pourra alors éventuellement naviguer parmi les dossiers de son disque dur. Au moment où il cliquera sur le bouton de sélection, le chemin d'accès sera mémorisé et associé au numéro 5. Lors des appels suivants du numéro 5, le chemin d'accès mémorisé sera utilisé par défaut. En cas de sélection d'un nouvel emplacement, le chemin numéro 5 sera mis à jour, et ainsi de suite.

Ce mécanisme vous permet de mémoriser jusqu'à 32000 chemins d'accès. Sous Windows, chaque chemin est conservé durant la session uniquement. Sous MacOS, les chemins sont conservés par le système, ils restent mémorisés d'une session à l'autre.

Passez dans le paramètre *typeFichiers* le ou les type(s) de fichier(s) pouvant être sélectionnés dans la boîte de dialogue d'ouverture. Vous pouvez passer une liste de plusieurs types séparés par un ; (point

virgule). Pour chaque type défini, une ligne sera ajoutée dans le menu de choix de type de la boîte de dialogue.

Sous MacOS, vous devez passer le type des fichiers pouvant être filtrés. Vous pouvez utiliser indifféremment le format OSType (TEXT, APPL, etc.) ou (à compter de MacOS 10.3) le format UTI (public.jpeg, public.tiff, etc.). Vous pouvez mélanger les deux formats dans le paramètre *typeFichiers*.

Sous Windows, vous pouvez également passer un type de fichier MacOS — 4<sup>e</sup> Dimension effectue la correspondance en interne — ou l’extension des fichiers (.txt, .exe, etc.). A noter que sous Windows, l’utilisateur aura la possibilité de “forcer” l’affichage de tous les types de fichiers en saisissant \*.\* dans la boîte de dialogue. Toutefois dans ce cas, 4<sup>e</sup> Dimension effectuera une vérification supplémentaire des types des fichiers sélectionnés : si l’utilisateur sélectionne un type de fichier non autorisé, la commande retourne une erreur.

Si vous ne souhaitez pas restreindre les fichiers affichés à un ou plusieurs types, passez la valeur \* (étoile) ou .\* dans *typeFichiers*.

Passez dans le paramètre *titre* le libellé devant apparaître dans la boîte de dialogue. Par défaut, si vous passez une chaîne vide, le libellé “Ouvrir” est affiché.

Le paramètre *options* permet de spécifier les fonctions avancées autorisées dans la boîte de dialogue d’ouverture. 4<sup>e</sup> Dimension vous propose les constantes prédéfinies suivantes, placées dans le thème Documents système :

Constante	Type	Valeur
Fichiers multiples	Entier long	1
Ouverture progiciel	Entier long	2
Sélection progiciel	Entier long	4
Sélection alias	Entier long	8
Utiliser fenêtre feuille	Entier long	16

Vous pouvez passer une constante ou une combinaison de constantes.

- Fichiers multiples : autoriser la sélection simultanée de plusieurs fichiers à l’aide des combinaisons **Maj+clik** (sélection contiguë) et **Ctrl+clik** (Windows) ou **Commande+clik** (MacOS). Dans ce cas, le paramètre *sélectionnés*, s’il est passé, contient la liste de tous les fichiers sélectionnés.

Par défaut, si cette constante n’est pas utilisée, la commande n’autorise pas la sélection de plusieurs fichiers.

- Ouverture progiciel (MacOS uniquement) : autoriser l'ouverture des progiciels (*packages*) et la visualisation de leur contenu.  
Par défaut, si cette constante n'est pas utilisée, la commande n'autorise pas l'ouverture des progiciels.
- Sélection progiciel (MacOS uniquement) : autoriser la sélection de progiciels (*packages*) en tant qu'entités.  
Par défaut, si cette constante n'est pas utilisée, la commande ne permet pas de sélectionner les progiciels. Dans ce cas, il est impossible d'ouvrir ou de sélectionner un progiciel (même si la constante Ouverture progiciel est passée).
- Sélection alias : autoriser la sélection de raccourcis (Windows) ou d'alias (MacOS) en tant que documents.  
Par défaut, si cette constante n'est pas utilisée, lorsqu'un alias ou un raccourci est sélectionné, la commande retourne le chemin d'accès de l'élément cible. Lorsque vous passez cette constante, la commande retourne le chemin de l'alias ou du raccourci lui-même.
- Utiliser fenêtre feuille (MacOS uniquement) : afficher la boîte de dialogue de sélection sous forme de fenêtre feuille (cette option est ignorée sous Windows).  
Les fenêtres feuilles sont des fenêtres spécifiques de l'interface MacOS X, bénéficiant notamment d'une animation graphique (pour plus d'informations, reportez-vous au [paragraphe "Type Fenêtre feuille", page 287](#)).  
Par défaut, si cette constante n'est pas utilisée, la commande affiche une boîte de dialogue standard.

Si vous ne souhaitez pas utiliser d'option, passez 0 dans le paramètre *options*.

Le paramètre facultatif *sélectionnés* permet de récupérer le chemin d'accès complet (chemin d'accès + nom) de chaque fichier sélectionné par l'utilisateur. La commande crée, dimensionne et remplit le tableau en fonction de la sélection de l'utilisateur. Ce paramètre est utile lorsque l'option Fichiers multiples est utilisée, ou lorsque vous souhaitez connaître le chemin d'accès du fichier sélectionné (il suffit dans ce cas de soustraire de la valeur du tableau le nom du fichier retourné par la commande).

Si aucun fichier n'a été sélectionné, le tableau est retourné vide.

La commande retourne le nom (nom+extension sous Windows) du fichier sélectionné.

Si plusieurs fichiers ont été sélectionnés, la commande retourne le nom



du premier fichier de la liste des fichiers sélectionnés. La liste des fichiers peut être récupérée dans le paramètre facultatif *sélectionnés*. Si aucun fichier n'a été sélectionné, la commande retourne une chaîne vide.

Si la commande a été correctement exécutée et qu'un document valide a été sélectionné, la variable système *OK* prend la valeur 1 et la variable système *Document* contient le chemin d'accès complet du fichier sélectionné.

Si aucun fichier n'a été sélectionné (par exemple si l'utilisateur a cliqué sur le bouton **Annuler** dans la boîte de dialogue d'ouverture), la variable système *OK* prend la valeur 0 et la variable système *Document* est vide.

## Propriétés des objets

Le thème "Propriétés des objets" s'est enrichi d'une nouvelle commande, **Lire formatage**, permettant de connaître le format d'affichage courant de tout objet.

En outre, les capacités des commandes existantes **CHOIX SAISSABLE**, **CHOIX FORMATAGE**, **FIXER COULEURS RVB** et **CHOIX COULEUR** ont été étendues.

### Lire formatage

Lire formatage ({\*; } objet) → Chaîne

Paramètres	Type	Description
*	*	→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est un champ ou une variable
objet	Objet de formulaire	→ Nom d'objet (si * spécifié) ou Champ ou variable (si * omis)
Résultat	Chaîne	← Format d'affichage de l'objet

La commande Lire formatage retourne le format d'affichage courant appliqué à l'objet spécifié par le paramètre *objet*.

Si vous passez le paramètre optionnel \*, vous indiquez que le paramètre *objet* est un nom d'objet (dans ce cas, passez une chaîne dans *objet*). Si vous ne passez pas le paramètre, vous indiquez que le

paramètre *objet* est un champ ou une variable. Dans ce cas, vous ne passez pas une chaîne mais une référence de champ ou de variable.

Cette commande retourne le format d'affichage courant de l'objet, c'est-à-dire le format défini en mode Structure ou à l'aide de la commande CHOIX FORMATAGE. Lire formatage fonctionne avec tous les types d'objets de formulaire (champs ou variables) acceptant un format d'affichage : booléen, date, heure, image, chaîne, numérique, ainsi que les grilles de boutons, cadrans, thermomètres, règles, pop up menus image, boutons image et boutons 3D. Pour plus d'informations sur les formats d'affichage de ces objets, reportez-vous à la documentation de la commande CHOIX FORMATAGE.

---

*Note* Si vous appliquez la commande à un ensemble d'objets, seul le formatage du dernier objet pris en compte est retourné.

---

Lorsque la commande Lire formatage est appliquée à des objets de type **date**, **heure** ou **image** (formats définis sous forme de constantes), la chaîne retournée correspond au code ASCII de la constante. Pour obtenir la valeur de la constante, il suffit d'appliquer la fonction Code ascii au résultat (cf. exemple ci-dessous).

- ▼ Cet exemple permet d'obtenir la valeur de la constante de formatage appliquée à la variable image dont le nom d'objet est "maphoto" :

```
C_ALPHA(2;$format)
```

```
CHOIX FORMATAGE(*;"maphoto";Caractere(Sur fond))
```

```
  `Application du format sur fond (valeur = 3)
```

```
$format:=Lire formatage(*;"maphoto")
```

```
ALERTE("Format numéro :"+Chaine(Code ascii($format)))
```

```
  `Affichage de la valeur "3"
```

- ▼ Cet exemple permet d'obtenir le formatage appliqué au champ booléen [Adhérents]Etat\_civil :

```
C_ALPHA(30;$format)
```

```
$format:=Lire formatage([Adhérents]Etat_civil)
```

```
ALERTE($format)
```

```
  `Affichage du format, par exemple "Marié;Célibataire"
```

**CHOIX SAISSABLE** CHOIX SAISSABLE({\*; }objet; zoneSaisie)

Paramètres	Type	Description
*		→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est <i>une table</i> , un champ ou une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou <i>Table</i> , champ ou variable (si * est omis)
zoneSaisie	Booléen	→ Vrai=saisissable ; Faux=non saisissable

La commande CHOIX SAISSABLE permet désormais d'activer par programmation le mode "Saisie en liste" pour les formulaires liste affichés via les commandes MODIFIER SELECTION et VISUALISER SELECTION et les sous-formulaires.

- Pour les formulaires liste, vous devez passer le nom de la table du formulaire dans le paramètre *objet*, par exemple : CHOIX SAISSABLE([MaTable];Vrai).
- Pour les sous-formulaires, vous pouvez passer dans le paramètre *objet* soit le nom de la table du sous-formulaire, soit le nom de l'objet sous-formulaire lui-même, par exemple : CHOIX SAISSABLE(\*;"Sousform";Vrai).

---

*Note* La syntaxe des commandes **MODIFIER SELECTION**, **VISUALISER SELECTION** a également été modifiée dans le but de permettre un meilleur contrôle des formulaires liste par programmation.

---

- ▼ Méthode objet d'une case à cocher placée dans l'en-tête d'une liste pour contrôler le mode Saisie en liste :

**C\_BOOLEEN**(bSaisissable)  
**CHOIX SAISSABLE**([Table1];bSaisissable)

**CHOIX FORMATAGE** CHOIX FORMATAGE ({\*;} objet; formatAffich)

Paramètres	Type	Description
*		→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est un champ ou une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Champ ou variable (si * est omis)
formatAffich	Numérique	→ Nouveau format d'affichage de l'objet

La commande CHOIX FORMATAGE (ainsi que la commande inverse [Lire formatage](#)) comporte des dispositions spécifiques pour l'affichage des boutons 3D (objets de type Bouton 3D, Bouton radio 3D et Case à cocher 3D).

*Note* Les boutons 3D acceptent de nombreuses nouvelles options dans 4<sup>e</sup> Dimension 2004, cf. [paragraphe "Modification des boutons 3D"](#), page 155.

**Boutons 3D**

Pour formater des boutons 3D, passez dans le paramètre *formatAffich* une chaîne de caractères respectant la syntaxe suivante :

titre;image;imageFond;posTitre;titreVisible;icôneVisible;style;margeHor; margeVert;décalageIcône;popupMenu

- titre = titre du bouton  
Cette valeur peut être exprimée sous forme de texte ou de numéro de ressource (ex. : ":16800,1")
- image = image associée au bouton, provenant de la bibliothèque d'images, d'une variable image ou d'une ressource PICT :
  - si l'image provient de la bibliothèque d'images, saisissez son numéro, précédé d'un point d'interrogation (ex. : "?250").
  - si l'image provient d'une variable image, saisissez le nom de la variable.
  - si l'image provient d'une ressource PICT, saisissez son numéro, précédé de deux points (ex. : ":62500").
- imageFond = image de fond associée au bouton (style Personnalisé), provenant de la bibliothèque d'images, d'une variable image ou d'une ressource PICT (cf. ci-dessus).
- posTitre = position du titre du bouton. Cinq valeurs sont possibles :
  - posTitre = 0 : Centre
  - posTitre = 1 : Droite

- posTitre = 2 : Gauche
- posTitre = 3 : Bas
- posTitre = 4 : Haut
- titreVisible = Titre visible ou non. Deux valeurs sont possibles :
  - titreVisible = 0 : le titre est masqué
  - titreVisible = 1 : le titre est affiché
- icôneVisible = Icône visible ou non. Deux valeurs sont possibles :
  - icôneVisible = 0 : l'icône est masquée
  - icôneVisible = 1 : l'icône est affichée
- style = Style du bouton. La valeur de cette option détermine la prise en compte de certaines autres options (par exemple imageFond). Dix valeurs de style sont possibles :
  - style = 0 : Aucun
  - style = 1 : Décalage du fond
  - style = 2 : Bouton poussoir
  - style = 3 : Bouton barre outils
  - style = 4 : Personnalisé
  - style = 5 : Rond
  - style = 6 : Petit carré système
  - style = 7 : Office XP
  - style = 8 : Bevel
  - style = 9 : Bevel arrondi
- margeHor = Marge horizontale  
Nombre de pixels délimitant les marges internes à droite et à gauche du bouton (zones dans lesquelles l'icône et le texte ne doivent pas empiéter).
- margeVert = Marge verticale  
Nombre de pixels délimitant les marges internes en haut et en bas du bouton (zones dans lesquelles l'icône et le texte ne doivent pas empiéter).
- décalageIcône = Décalage de l'icône vers la droite et le bas  
Cette valeur, exprimée en pixels, indique le décalage de l'icône du bouton vers la droite et le bas en cas de clic (la même valeur est utilisée pour les deux directions).
- popupMenu = Association d'un pop up menu au bouton. Trois valeurs sont possibles :
  - popupMenu = 0 : Sans pop up menu
  - popupMenu = 1 : Avec pop up menu lié
  - popupMenu = 2 : Avec pop up menu séparé

Certaines options ne sont pas prises en charge dans tous les styles de boutons 3D. De plus, dans certains cas vous pourrez souhaiter ne pas modifier toutes les options.

Pour ne pas passer une option, il suffit d'omettre la valeur correspondante. Par exemple, pour ne pas passer les options `titreVisible` et `margeVert`, vous pouvez écrire :

**CHOIX FORMATAGE**(`maVar;"JoliBouton";?256;;562;1;;1;4;5;;5;0"`)

## FIXER COULEURS RVB

FIXER COULEURS RVB(`{*; }objet; couleurAvantPlan; couleurArrièrePlan{; couleurArrièrePlanAlt}`)

Paramètres	Type	Description
*		→ Si spécifié, <code>objet</code> est un nom d'objet (chaîne) Si omis, <code>objet</code> est un champ ou une variable
<code>objet</code>	Objet	→ Nom d'objet (si * est spécifié) ou Champ ou Variable (si * est omis)
<code>couleurAvantPlan</code>	Numérique	→ Valeur de la couleur RVB d'avant-plan
<code>couleurArrièrePlan</code>	Numérique	→ Valeur de la couleur RVB d'arrière-plan
<code>couleurArrièrePlanAlt</code>	Numérique	→ Valeur de la couleur RVB d'arrière-plan alternée

La commande `FIXER COULEURS RVB` présente deux nouveautés dans 4<sup>e</sup> Dimension 2004 :

- le nouveau paramètre facultatif `couleurArrièrePlanAlt`,
- de nouvelles constantes de couleurs automatiques.

### `couleurArrièrePlanAlt`

La commande `FIXER COULEURS RVB` accepte un nouveau paramètre facultatif, `couleurArrièrePlanAlt`. Ce paramètre n'est utile que lorsque l'*objet* désigné est de type list box ou colonne de list box (pour plus d'informations sur les list box, reportez-vous au [paragraphe "List box", page 328](#)).

`couleurArrièrePlanAlt` permet de désigner une couleur alternative pour l'arrière-plan (c'est-à-dire le fond) des lignes paires. Lorsque ce paramètre est utilisé, la `couleurArrièrePlan` est utilisée pour le fond des lignes impaires uniquement. Utiliser des couleurs alternées améliore la lisibilité des tableaux.

Si *objet* désigne l'objet list box, les couleurs alternées sont utilisées dans la totalité de la list box. Si *objet* désigne une colonne, seule la colonne utilisera les couleurs définies.

### Constantes de couleur

La commande FIXER COULEURS RVB admet de nouvelles constantes de couleurs système par défaut dans les paramètres *couleurAvantPlan*, *couleurArrièrePlan* et *couleurArrièrePlanAlt*. A noter également que pour des raisons de cohérence, les quatre constantes initiales du thème de constantes "FIXER COULEUR RVB" ont été renommées.

Constante ( <i>ancien nom</i> )	Type	Valeur
Coul premier plan ( <i>Couleur par défaut premier plan</i> )	Entier long	-1
Coul arrière plan ( <i>Couleur par défaut arrière plan</i> )	Entier long	-2
Coul sombre ( <i>Couleur sombre par défaut</i> )	Entier long	-3
Coul claire ( <i>Couleur claire par défaut</i> )	Entier long	-4
Coul de fond texte sélect	Entier long	-7
Coul texte sélect	Entier long	-8
Coul fond ligne menu sélect	Entier long	-9
Coul texte ligne menu sélect	Entier long	-10
Coul fond élément sélect désact	Entier long	-11

### CHOIX COULEUR

CHOIX COULEUR({\*; }objet; couleur{; couleurAlt})

Paramètres	Type	Description
*		→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est un champ ou une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Champ ou Variable (si * est omis)
couleur	Numérique	→ Nouvelles couleurs pour l'objet
<i>couleurAlt</i>	<i>Numérique</i>	→ <i>Couleurs alternées pour une list box</i>

La commande CHOIX COULEUR accepte un paramètre facultatif supplémentaire, *couleurAlt*. Ce paramètre n'est utile que lorsque l'*objet* désigné est de type list box ou colonne de list box.

*couleurAlt* permet de désigner une couleur alternative pour les lignes paires. Lorsque ce paramètre est passé, la *couleur* s'applique aux lignes impaires uniquement. Utiliser des couleurs alternées améliore la lisibilité des tableaux.

Si *objet* désigne l'objet list box, les couleurs alternées sont utilisées dans la totalité de la list box. Si *objet* désigne une colonne, seule la colonne utilisera les couleurs définies.

## List box

Les commandes de ce nouveau thème sont dédiées à la gestion des nouveaux types d'objets de formulaire, les *list box*.

Les list box peuvent être comparés aux "zones de défilement groupées" des versions précédentes de 4<sup>e</sup> Dimension. De fait, une list box propose toutes les fonctions des zones de défilement groupées, notamment la possibilité de représenter des données sous forme de colonnes et de lignes sélectionnables. Mais la list box de 4<sup>e</sup> Dimension 2004 propose de nombreuses fonctions supplémentaires telles que la possibilité de saisir des valeurs, de trier les colonnes, de définir des couleurs alternées, etc. Un objet de type List box est entièrement paramétrable dans l'éditeur de formulaires de 4<sup>e</sup> Dimension et peut également être contrôlé par programmation.

Ce paragraphe détaille les nouveautés liées à la gestion programmée des objets de type List box. Pour plus d'informations sur la création et le paramétrage des objets de type List box, reportez-vous au [paragraphe "List Box", page 133](#).

Ce paragraphe décrit :

- les principes de programmation des objets de type List box,
- les nouvelles commandes de gestion de cet objet : INSERER COLONNE LISTBOX, SUPPRIMER COLONNE LISTBOX, Lire nombre colonnes listbox, TRIER COLONNES LISTBOX, FIXER LARGEUR COLONNE LISTBOX, Lire largeur colonne listbox, SELECTIONNER LIGNE LISTBOX, INSERER LIGNE LISTBOX, SUPPRIMER LIGNE LISTBOX, Lire nombre lignes listbox, FIXER HAUTEUR LIGNES LISTBOX, Lire hauteur lignes listbox, NUMERO LIGNE LISTBOX DEPLACEE, LIRE TABLEAUX LISTBOX, Lire information listbox, MONTRER GRILLE LISTBOX, FIXER COULEUR GRILLE LISTBOX, MONTRER BARRES DEFILEMENT LISTBOX et NUMERO COLONNE LISTBOX DEPLACEE.

Les objets de type List box ont également entraîné la modification des commandes CHOIX COULEUR et FIXER COULEURS RVB, décrites dans le [paragraphe "Propriétés des objets", page 321](#).



## Principes de programmation des objets de type List box

La programmation des objets de type “List box” s’effectue sur le même principe que les autres objets de formulaire en liste de 4<sup>e</sup> Dimension. Elle doit cependant tenir compte de principes spécifiques, décrits dans ce paragraphe.

## Création et gestion des valeurs

Un objet “List box” peut contenir une ou plusieurs colonnes. Chaque colonne doit être associée à un tableau 4<sup>e</sup> Dimension à une dimension ; tous les types de tableaux peuvent être utilisés, à l’exception des tableaux de pointeurs. Le format d’affichage de chaque colonne peut être défini dans l’éditeur de formulaires ou via la commande CHOIX FORMATAGE.

En mode programmé, les valeurs des colonnes (saisie et affichage) sont gérées à l’aide des commandes de haut niveau du thème List box (telles que [INSERER LIGNE LISTBOX](#) ou [SUPPRIMER LIGNE LISTBOX](#)), des commandes du thème “Propriétés des objets” (telle que la commande CHOIX SAISSABLE) et des commandes de manipulation des tableaux. Par exemple, pour initialiser le contenu d’une colonne, vous pouvez utiliser l’instruction suivante :

**TABLEAU TEXTE**(NomColonne; taille)

Vous pouvez également utiliser une énumération :

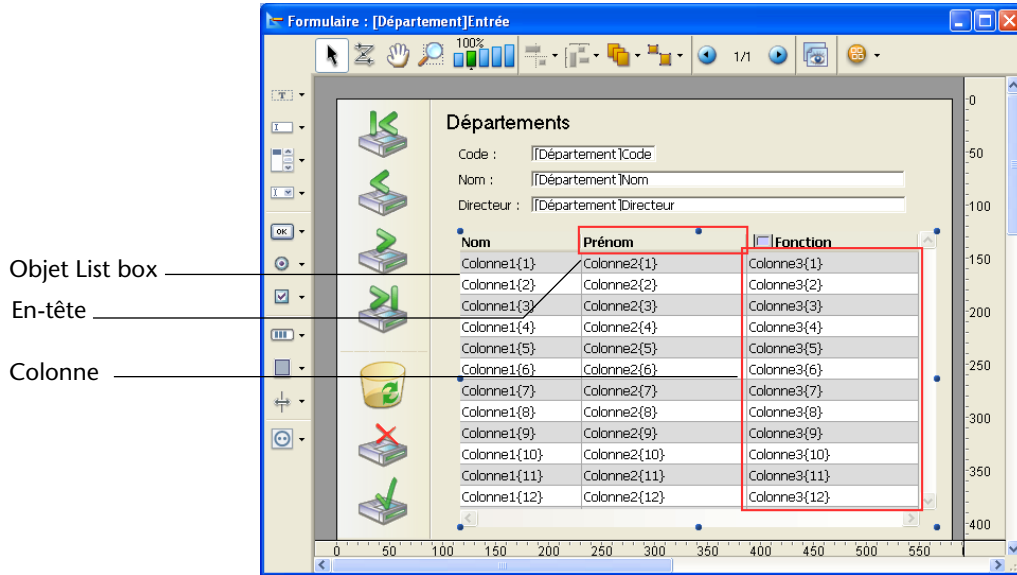
**ENUMERATION VERS TABLEAU**(NomEnum; NomColonne)

- 
- Notes*
- Lorsqu’un objet List box contient plusieurs colonnes, chaque tableau associé doit avoir la même taille (le même nombre d’éléments) que les autres, sinon seul le nombre d’éléments correspondant au plus petit tableau sera affiché.
  - Deux nouvelles commandes ont été ajoutées dans le thème “Tableaux” dans 4<sup>e</sup> Dimension 2004 : [Compter dans tableau](#) et [AJOUTER A TABLEAU](#).
- 

## Objet, colonne et en-tête

Un objet List box est composé de trois types d’éléments distincts : l’**objet** dans son ensemble, les **colonnes** et les **en-têtes** des colonnes. Dans l’éditeur de formulaires, ces éléments peuvent être sélectionnés

séparément. Chacun d’eux dispose de son propre nom d’objet et de variable et peut donc être adressé séparément.



Par défaut, les colonnes sont nommées *Colonne1* à *n* et les en-têtes *Entête1* à *n* dans le formulaire, indépendamment des objets list box eux-mêmes. A noter que, par défaut, le même nom est utilisé pour les objets et leurs variables associées.

Chaque type d’élément dispose de caractéristiques propres et de caractéristiques partagées avec les autres éléments. Par exemple, la police de caractères peut être assignée globalement à l’objet list box ou séparément aux colonnes et aux en-têtes. A l’inverse, les propriétés de saisie ne sont définissables que pour les colonnes.

Ces principes s’appliquent aux commandes du thème “Propriétés des objets” pouvant être utilisées avec les list box : en fonction de sa nature, chaque commande sera utilisable avec la list box, les colonnes et/ou les en-têtes des colonnes. Pour désigner le type d’élément sur lequel vous souhaitez agir, il suffit de passer le nom ou la variable qui lui est associé(e).

Le tableau suivant précise la portée de chaque commande du thème “Propriétés des objets” utilisable avec les objets de type list box :

<i>Commandes du thème Propriétés des objets</i>	<b>Objet</b>	<b>Colonne</b>	<b>En-tête de colonne</b>
DEPLACER OBJET	X		
LIRE RECT OBJET	X		
CHOIX FILTRE SAISIE		X	
CHOIX FORMATAGE		X	
CHOIX SAISSABLE		X	
CHOIX ENUMERATION		X	
TITRE BOUTON			X
CHOIX COULEUR	X	X	
FIXER COULEURS RVB	X	X	
CHANGER JEU DE CARACTERES	X	X	X
CHANGER TAILLE	X	X	X
CHANGER STYLE	X	X	X
FIXER ALIGNEMENT	X	X	X
Lire alignement	X	X	X
CHOIX VISIBLE	X	X	X
TAILLE OBJET OPTIMALE	X	X	X

*Note* Toutes les commandes du nouveau thème “List Box” s’appliquent à l’objet List box uniquement, à l’exception de **FIXER LARGEUR COLONNE LISTBOX** (objet, colonne et en-tête) et **Lire largeur colonne listbox** (colonne ou en-tête uniquement).

## Méthodes objet

Il est possible d’associer une méthode à l’objet List box dans son ensemble et/ou à chaque colonne de la list box. Les méthodes objet sont appelées dans l’ordre suivant :

- 1. Méthode objet de la colonne
- 2. Méthode objet de la list box

La méthode objet de la colonne reçoit les événements se produisant dans son en-tête.

### CHOIX VISIBLE et en-têtes

Lorsque la commande CHOIX VISIBLE est utilisée avec un en-tête, elle agit sur tous les en-têtes de l'objet List box, quel que soit l'en-tête spécifié par la commande. Par exemple, l'instruction CHOIX VISIBLE(\*;"entête3";Faux) masquera tous les en-têtes de l'objet List box auquel appartient l'en-tête 3 (et non uniquement cet en-tête).

### Self

La fonction Self (thème "Langage") peut être utilisée dans la méthode objet d'une list box ou d'une colonne de list box. Dans les deux cas, elle retourne un pointeur vers la variable de colonne ou la variable d'en-tête en fonction de l'emplacement du clic.

### Objet focus

**Note préliminaire** : Objet focus est le nouveau nom de la commande existante Dernier objet (thème "Interface utilisateur"). Cette commande a été renommée par souci de clarté.

La fonction Objet focus retourne un pointeur vers la colonne de la list box ayant le focus, c'est-à-dire vers un tableau. Le mécanisme des pointeurs de 4<sup>e</sup> Dimension permet alors de connaître le numéro de l'élément de tableau modifié. Par exemple, en supposant que l'utilisateur a modifié la 5e ligne de la colonne *col2* :

```
$Colonne:=Objet focus  
  ` $Colonne contient un pointeur vers col2  
$Ligne:= $Colonne-> ` $Ligne vaut 5
```

### EDITER ELEMENT

La nouvelle commande EDITER ELEMENT (thème "Gestion de la saisie") permet de passer en mode édition une cellule d'un objet list box. Pour plus d'informations sur ce point, reportez-vous à la description de la [commande EDITER ELEMENT, page 312](#).

### Evénements formulaire

De nouveaux événements formulaire ont été créés pour faciliter la gestion programmée des list box, concernant notamment le glisser-déposer et le tri. Pour plus d'informations, reportez-vous au [paragraphe "Evénements formulaire liés aux list box", page 304](#).

### Variable d'en-tête et gestion des tris

Par défaut, la list box gère automatiquement les tris standard des colonnes en cas de clic sur l'en-tête. Un tri standard est un tri alphanumérique des valeurs de la colonne, alternativement croissant / décroissant lors de clics multiples. Toutes les colonnes sont toujours automatiquement synchronisées.

Il est possible d'interdire le tri utilisateur standard en désélectionnant la propriété "Triable" pour la list box.

Le développeur peut mettre en place des tris personnalisés à l'aide de la commande **TRIER COLONNES LISTBOX** et/ou en combinant les événements formulaire **Sur clic entête** et **Sur après tri** (cf. [paragraphe "Événements formulaire liés aux list box"](#), page 304) et les commandes 4D de gestion des tableaux.

*Note* La propriété "Triable" concerne uniquement le tri utilisateur standard, la commande **TRIER COLONNES LISTBOX** ne tient pas compte de cette propriété.

La valeur de la variable associée à l'en-tête d'une colonne permet de gérer une information supplémentaire : le tri courant de la colonne (lecture) et l'affichage de la flèche de tri.

- si la variable vaut 0, la colonne n'est pas triée et la flèche de tri n'est pas affichée ;

colonne2

- si la variable vaut 1, la colonne est triée par ordre croissant et la flèche de tri croissant est affichée ;

colonne2 ▲

- si la variable vaut 2, la colonne est triée par ordre décroissant et la flèche de tri décroissant est affichée.

colonne2 ▼

Il est possible de fixer la valeur de la variable (par exemple Header2:=2) afin de "forcer" l'affichage de la flèche de tri. Le tri de la colonne lui-même n'est dans ce cas pas modifié, il appartient au développeur de le gérer.

## Variable d'objet et gestion des sélections

La variable associée à l'objet List box est utilisée pour lire, fixer ou stocker les sélections de lignes dans l'objet.

Cette variable correspond à un tableau de booléens automatiquement créé et maintenu par 4<sup>e</sup> Dimension. La taille de ce tableau est déterminée par celle de la list box : il contient le même nombre d'éléments que le plus petit tableau associé aux colonnes.

Chaque élément de ce tableau contient Vrai si la ligne correspondante est sélectionnée et Faux sinon. 4<sup>e</sup> Dimension met à jour le contenu de ce tableau en fonction des actions utilisateur. A l'inverse, vous pouvez modifier la valeur des éléments de ce tableau afin de modifier la sélection dans la list box.

En revanche, vous ne pouvez ni insérer ni supprimer de ligne dans ce tableau ; il n'est pas possible non plus de le retyper.

*Note* La nouvelle commande **Compter dans tableau** est utile pour connaître le nombre de lignes sélectionnées.

- ▼ Cette méthode permet d'inverser la sélection de la première ligne de la list box :

  ` tListBox est le nom de la variable de la List box

**Si** (tListBox{1} = **Vrai**)

  tListBox{1}:= **Faux**

**Sinon**

  tListBox{1}:= **Vrai**

**Fin de si**

## INSERER COLONNE LISTBOX

INSERER COLONNE LISTBOX({\*; }objet; position; nom; variable; nomEntête; variableEntête)

Paramètres	Type	Description
*		→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
position	Numérique	→ Emplacement de la colonne à insérer
nom	Alpha	→ Nom d'objet de la colonne
variable	Tableau	→ Nom de la variable de la colonne
nomEntête	Alpha	→ Nom d'objet de l'en-tête de la colonne
variableEntête	Var Entier	→ Variable d'en-tête de la colonne

La commande INSERER COLONNE LISTBOX insère une colonne dans la list box désignée par les paramètres *objet* et *\**.

Si vous passez le paramètre optionnel *\**, vous indiquez que le paramètre *objet* est un nom d'objet (une chaîne). Si vous ne passez pas ce paramètre, vous indiquez que le paramètre *objet* est une variable. Dans ce cas, vous ne passez pas une chaîne mais une référence de variable.

La nouvelle colonne est insérée juste avant la colonne désignée par le paramètre *position*. Si le paramètre *position* est supérieur au nombre total de colonnes, la colonne est ajoutée après la dernière colonne.

Passez dans les paramètres *nom* et *variable* le nom d'objet et la variable de la colonne insérée. Le nom de la variable doit correspondre au nom du tableau dont le contenu sera affiché dans la colonne.

Passez dans les paramètres *nomEntête* et *variableEntête* le nom d'objet et la variable de l'en-tête de la colonne insérée.

*Note* Les noms d'objets doivent être uniques dans un formulaire. Vous devez veiller à ce que les noms passés dans les paramètres *nom* et *nomEntête* ne soient pas déjà utilisés. Sinon, la colonne n'est pas créée et une erreur est générée.

- ▼ Nous souhaitons ajouter une colonne à la fin de la list box :

```
C_ENTIER LONG(NomVarHeader;$Der;$NbEnr)
```

```
TOUT SELECTIONNER([Table 1])
```

```
$NbEnr:=Enregistrements dans table([Table 1])
```

```
TABLEAU IMAGE(Picture;$NbEnr)
```

```
$Der:=Lire nombre colonnes listbox(*;"ListBox1")+1
```

```
INSERER COLONNE LISTBOX(*;"ListBox1";$Der;"ColumnPicture";Picture;  
"HeaderPicture";NomVarHeader)
```

## SUPPRIMER COLONNE LISTBOX

```
SUPPRIMER COLONNE LISTBOX({*; }objet; position{; nombre})
```

Paramètres	Type	Description
*		→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
position	Numérique	→ Numéro de la colonne à supprimer
nombre	Numérique	→ Nombre de colonnes à supprimer

La commande SUPPRIMER COLONNE LISTBOX supprime une ou plusieurs colonne(s) (visibles ou non) dans la list box désignée par les paramètres *objet* et *\**.

Si vous passez le paramètre optionnel *\**, vous indiquez que le paramètre *objet* est un nom d'objet (une chaîne). Si vous ne passez pas ce paramètre, vous indiquez que le paramètre *objet* est une variable. Dans ce cas, vous ne passez pas une chaîne mais une référence de variable.

Si vous ne passez pas le paramètre facultatif *nombre*, la commande supprime simplement la colonne désignée par le paramètre *position*. Sinon, le paramètre *nombre* indique le nombre de colonnes à supprimer vers la droite à partir de la colonne *position* (celle-ci incluse).

Si le paramètre *position* est supérieur au nombre de colonnes de la list box, la commande ne fait rien.

## Lire nombre colonnes listbox

Lire nombre colonnes listbox ({\*; }objet) → Entier long

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

La commande Lire nombre colonnes listbox retourne le nombre total de colonnes (visibles ou non) présentes dans la list box désignée par les paramètres *objet* et *\**.

Si vous passez le paramètre optionnel *\**, vous indiquez que le paramètre *objet* est un nom d'objet (une chaîne). Si vous ne passez pas ce paramètre, vous indiquez que le paramètre *objet* est une variable. Dans ce cas, vous ne passez pas une chaîne mais une référence de variable.

## TRIER COLONNES LISTBOX

TRIER COLONNES LISTBOX({\*; }objet; numCol; sensDuTri{; numCol2; sensDuTri2}...{; numColN; sensDuTriN)

Paramètres	Type	Description
*		→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
numCol	Entier long	→ Numéro(s) de colonne(s) de tri
sensDuTri1...N	> ou <	→ > pour effectuer un tri croissant ou < pour effectuer un tri décroissant



La commande TRIER COLONNES LISTBOX permet de trier (réordonner) toutes les lignes de la list box désignée par les paramètres *objet* et *\** sur la base des valeurs d'une ou plusieurs colonne(s).

Si vous passez le paramètre optionnel *\**, vous indiquez que le paramètre *objet* est un nom d'objet (une chaîne). Si vous ne passez pas ce paramètre, vous indiquez que le paramètre *objet* est une variable. Dans ce cas, vous ne passez pas une chaîne mais une référence de variable.

Passez dans *numCol* le numéro de la colonne dont les valeurs seront utilisées comme critère de tri. Vous pouvez utiliser tout type de données, à l'exception des colonnes contenant des images ou des pointeurs.

Passez dans *sensDuTri* le symbole > ou < indiquant l'ordre du tri : croissant ou décroissant. Si *sensDuTri* est égal au symbole "supérieur à" (>), l'ordre du tri est croissant. Si *sensDuTri* est égal au symbole "inférieur à" (<), l'ordre du tri est décroissant.

Vous pouvez définir des tris multi-niveaux : pour cela, passez autant de paires *numCol*;*sensDuTri* que nécessaire. Le niveau de tri est défini par la position du paramètre lors de l'appel.

Conformément au principe de fonctionnement des list box, les colonnes sont synchronisées, ce qui signifie que le tri d'une colonne est automatiquement répercuté sur toutes les colonnes de l'objet.

## FIXER LARGEUR COLONNE LISTBOX

FIXER LARGEUR COLONNE LISTBOX ({\*; }objet; largeur)

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

La commande FIXER LARGEUR COLONNE LISTBOX permet de modifier par programmation la largeur d'une ou de toutes les colonne(s) de l'objet (list box, colonne ou en-tête) désigné par les paramètres *objet* et *\**.

Si vous passez le paramètre optionnel *\**, vous indiquez que le paramètre *objet* est un nom d'objet (une chaîne). Si vous ne passez pas ce paramètre, vous indiquez que le paramètre *objet* est une variable. Dans ce cas, vous ne passez pas une chaîne mais une référence de variable.

Passez dans le paramètre *largeur* la nouvelle largeur (en pixels) de l'*objet*.

- Si *objet* désigne l'objet list box, toutes les colonnes de la list box sont redimensionnées.
- Si *objet* désigne une colonne ou un en-tête de colonne, seule la colonne désignée est redimensionnée.

### Lire largeur colonne listbox

Lire largeur colonne listbox ({\*; }objet) → Entier

Paramètres	Type	Description
<i>*</i>		→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si <i>*</i> est spécifié) ou Variable (si <i>*</i> est omis)
Résultat	Entier	← Largeur de colonne en pixels

La commande Lire largeur colonne listbox retourne la largeur (en pixels) de la colonne désignée par les paramètres *objet* et *\**.

Vous pouvez passer indifféremment une colonne ou un en-tête de colonne de list box dans le paramètre *objet*.

Si vous passez le paramètre optionnel *\**, vous indiquez que le paramètre *objet* est un nom d'objet (une chaîne). Si vous ne passez pas ce paramètre, vous indiquez que le paramètre *objet* est une variable. Dans ce cas, vous ne passez pas une chaîne mais une référence de variable.

**NUMERO COLONNE  
LISTBOX DEPLACEE**

NUMERO COLONNE LISTBOX DEPLACEE ({\*; }objet; ancPosition; nouvPosition)

Paramètres	Type	Description
*		→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
ancPosition	Numérique	← Ancienne position de la colonne déplacée
nouvPosition	Numérique	← Nouvelle position de la colonne déplacée

La commande NUMERO COLONNE LISTBOX DEPLACEE retourne dans les paramètres *ancPosition* et *nouvPosition* des numéros indiquant respectivement la précédente position et la nouvelle position de la colonne déplacée dans la list box désignée par les paramètres *objet* et *\**.

Cette commande doit être utilisée en combinaison avec l'événement formulaire Sur déplacement colonne (cf. [paragraphe "Événements formulaire liés aux list box", page 304](#)).

---

*Note* Cette commande tient compte des colonnes invisibles.

---

**SELECTIONNER  
LIGNE LISTBOX**

SELECTIONNER LIGNE LISTBOX({\*; }objet; position{; action})

Paramètres	Type	Description
*		→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
position	Entier long	→ Numéro de la ligne à sélectionner
action	Entier long	→ Action de sélection

La commande **SELECTIONNER LIGNE LISTBOX** provoque la sélection de la ligne de numéro *position* dans l'objet list box désigné par les paramètres *objet* et *\**.

Si vous passez le paramètre optionnel *\**, vous indiquez que le paramètre *objet* est un nom d'objet (une chaîne). Si vous ne passez pas ce paramètre, vous indiquez que le paramètre *objet* est une variable.

Dans ce cas, vous ne passez pas une chaîne mais une référence de variable.

Le paramètre *action*, s'il est passé, permet de définir l'action de sélection à effectuer lorsqu'une sélection de lignes existe déjà dans la list box. Vous pouvez passer une valeur ou l'une des constantes suivantes, placées dans le thème "List box" :

- Remplacer sélection listbox (0) : la ligne sélectionnée devient la nouvelle sélection et remplace la sélection existante. La commande produit le même effet qu'un clic de l'utilisateur sur une ligne de la list box. Cette action est effectuée par défaut (lorsque le paramètre *action* n'est pas passé).
- Ajouter à sélection listbox (1) : la ligne sélectionnée est ajoutée à la sélection existante. Si la ligne désignée appartient déjà à la sélection existante, la commande ne fait rien.
- Supprimer de sélection listbox (2) : la ligne sélectionnée est supprimée de la sélection existante. Si la ligne désignée n'appartient pas à la sélection existante, la commande ne fait rien.

## INSERER LIGNE LISTBOX

INSERER LIGNE LISTBOX({\*; }objet; position)

Paramètres	Type	Description
*		→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
position	Entier long	→ Emplacement de la ligne à insérer

La commande **INSERER LIGNE LISTBOX** insère une nouvelle ligne dans l'objet list box désigné par les paramètres *objet* et *\**.

Si vous passez le paramètre optionnel *\**, vous indiquez que le paramètre *objet* est un nom d'objet (une chaîne). Si vous ne passez pas ce paramètre, vous indiquez que le paramètre *objet* est une variable. Dans ce cas, vous ne passez pas une chaîne mais une référence de variable.

La ligne est insérée à l'emplacement défini par le paramètre *position*. Une nouvelle ligne est automatiquement ajoutée à cet emplacement dans tous les tableaux composant la list box, quel que soit leur type, y compris dans les tableaux (colonnes) masqués.

Si le paramètre *position* est supérieur au nombre de lignes des tableaux de la list box, la ligne est ajoutée à la fin de chaque tableau.

## SUPPRIMER LIGNE LISTBOX

SUPPRIMER LIGNE LISTBOX({\*; }objet; position)

Paramètres	Type	Description
*		→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
position	Entier long	→ Numéro de la ligne à supprimer

La commande **SUPPRIMER LIGNE LISTBOX** supprime la ligne numéro *position* (visible ou non) de la list box désignée par les paramètres *objet* et *\**.

Si vous passez le paramètre optionnel *\**, vous indiquez que le paramètre *objet* est un nom d'objet (une chaîne). Si vous ne passez pas ce paramètre, vous indiquez que ce paramètre est une variable. Dans ce cas, vous ne passez pas une chaîne mais une référence de variable.

La ligne *position* est supprimée automatiquement dans tous les tableaux composant la list box.

Notez qu'après l'exécution de la commande, il n'y a plus d'élément sélectionné dans la list box.

Si le paramètre *position* est supérieur au nombre de lignes des tableaux de la list box, la commande ne fait rien.

## Lire nombre lignes listbox

Lire nombre lignes listbox ({\*; }objet) → Entier long

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

La commande Lire nombre lignes listbox retourne le nombre de lignes présentes dans la list box désignée par les paramètres *objet* et *\**.

Si vous passez le paramètre optionnel \*, vous indiquez que le paramètre *objet* est un nom d'objet (une chaîne). Si vous ne passez pas ce paramètre, vous indiquez que le paramètre *objet* est une variable. Dans ce cas, vous ne passez pas une chaîne mais une référence de variable.

## FIXER HAUTEUR LIGNES LISTBOX

FIXER HAUTEUR LIGNES LISTBOX {\*; }objet; hauteur)

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

La commande FIXER HAUTEUR LIGNES LISTBOX permet de modifier par programmation la hauteur des lignes de l'objet list box désigné par les paramètres *objet* et \*.

Si vous passez le paramètre optionnel \*, vous indiquez que le paramètre *objet* est un nom d'objet (une chaîne). Si vous ne passez pas ce paramètre, vous indiquez que le paramètre *objet* est une variable. Dans ce cas, vous ne passez pas une chaîne mais une référence de variable.

## Lire hauteur lignes listbox

Lire hauteur lignes listbox ({\*; }objet) → Entier

Paramètres	Type	Description
*		→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
Résultat	Entier	← Hauteur de ligne en pixels

La commande Lire hauteur lignes listbox retourne la hauteur courante (en pixels) d'une ligne de l'objet list box désigné par les paramètres *objet* et \*.

Si vous passez le paramètre optionnel *\**, vous indiquez que le paramètre *objet* est un nom d'objet (une chaîne). Si vous ne passez pas ce paramètre, vous indiquez que le paramètre *objet* est une variable. Dans ce cas, vous ne passez pas une chaîne mais une référence de variable.

## NUMERO LIGNE LISTBOX DEPLACEE

NUMERO LIGNE LISTBOX DEPLACEE ({\*}; )objet; ancPosition; nouvPosition)

Paramètres	Type	Description
*		→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
ancPosition	Numérique	← Ancienne position de la ligne déplacée
nouvPosition	Numérique	← Nouvelle position de la ligne déplacée

La commande NUMERO LIGNE LISTBOX DEPLACEE retourne dans les paramètres *ancPosition* et *nouvPosition* des numéros indiquant respectivement la précédente position et la nouvelle position de la ligne déplacée dans la list box désignée par les paramètres *objet* et *\**.

Si vous passez le paramètre optionnel *\**, vous indiquez que le paramètre *objet* est un nom d'objet (une chaîne). Si vous ne passez pas ce paramètre, vous indiquez que le paramètre *objet* est une variable. Dans ce cas, vous ne passez pas une chaîne mais une référence de variable.

Cette commande doit être utilisée en combinaison avec l'événement formulaire Sur déplacement ligne (cf. [paragraphe "Evénements formulaire liés aux list box"](#), page 304).

## LIRE TABLEAUX LISTBOX

LIRE TABLEAUX LISTBOX({\*; }objet; tabNomsCols; tabNomsEntêtes; tabVariablesCols; tabVariablesEntêtes; tabVisibles; tabStyles)

Paramètres	Type	Description
*		→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
tabNomsCols	Tab alpha	← Noms d'objet des colonnes
tabNomsEntêtes	Tab alpha	← Noms d'objet des en-têtes
tabVariablesCols	Tab pointeur	← Pointeurs vers les variables des colonnes
tabVariablesEntêtes	Tab pointeur	← Pointeurs vers les variables des en-têtes
tabVisibles	Tab booléen	← Visibilité de chaque colonne
tabStyles	Tab pointeur	← Pointeurs vers les tableaux de styles et de couleurs

La commande LIRE TABLEAUX LISTBOX retourne un ensemble de tableaux synchronisés fournissant plusieurs informations sur chaque colonne (visible ou non) de la list box désignée par les paramètres *objet* et *\**.

Si vous passez le paramètre optionnel *\**, vous indiquez que le paramètre *objet* est un nom d'objet (une chaîne). Si vous ne passez pas ce paramètre, vous indiquez que le paramètre *objet* est une variable. Dans ce cas, vous ne passez pas une chaîne mais une référence de variable.

A l'issue de l'exécution de la commande :

- le tableau *tabNomsCols* contient la liste des noms d'objet de chaque colonne de la list box.
- le tableau *tabNomsEntêtes* contient la liste des noms d'objet de chaque en-tête de colonne de la list box.
- le tableau *tabVariablesCols* contient des pointeurs vers les variables (c'est-à-dire les tableaux) associées à chaque colonne de la list box.
- le tableau *tabVariablesEntêtes* contient des pointeurs vers les variables associées à chaque en-tête de colonne de la list box.



- le tableau *tabVisibles* contient une valeur booléenne pour chaque colonne, indiquant si la colonne est visible (valeur Vrai) ou masquée (valeur Faux) dans la list box.
- le tableau *tabStyles* contient trois pointeurs vers les trois tableaux permettant d'appliquer individuellement un style, une couleur de police et une couleur de fond personnalisées à chaque ligne du tableau. Ces tableaux sont associés à la list box dans la Liste des propriétés en mode Structure (pour plus d'informations, reportez-vous au [paragraphe "Propriétés spécifiques de l'objet List Box", page 137](#)). Si un tableau n'est pas spécifié pour la list box, l'élément correspondant de *tabStyles* contient un pointeur Nil.

## Lire information listbox

Lire information listbox ({\*; }objet; info) → Entier long

Paramètres	Type	Description
*		→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
info	Entier long	→ Type d'information à récupérer
Résultat	Entier long	← Valeur courante

La commande Lire information listbox permet de récupérer différentes informations concernant la taille et la visibilité des en-têtes et des barres de défilement de l'objet list box désigné par les paramètres *objet* et *\**.

Si vous passez le paramètre optionnel *\**, vous indiquez que le paramètre *objet* est un nom d'objet (une chaîne). Si vous ne passez pas ce paramètre, vous indiquez que le paramètre *objet* est une variable. Dans ce cas, vous ne passez pas une chaîne mais une référence de variable.

Passez dans le paramètre *info* un code correspondant à l'information que vous souhaitez obtenir.

Vous pouvez passer une valeur ou l'une des constantes suivantes, placées dans le thème "List box" :

Constante	Type	Valeur	Valeur(s) retournée(s)
Affichage entête listbox	Entier long	0	0=masqué, 1=affiché
Hauteur entête listbox	Entier long	1	Hauteur en pixels
Affichage barre déf hor listbox	Entier long	2	0=masqué, 1=affiché
Hauteur barre déf hor listbox	Entier long	3	Hauteur en pixels
Affichage barre déf ver listbox	Entier long	4	0=masqué, 1=affiché
Largeur barre déf ver listbox	Entier long	5	Largeur en pixels

Ces informations sont utiles pour déterminer avec précision la taille de la zone de list box affichée dans le formulaire.

## MONTRER GRILLE LISTBOX

MONTRER GRILLE LISTBOX({\*; }objet; horizontal; vertical)

Paramètres	Type	Description
*		→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
horizontal	Booléen	→ Vrai=montrer, Faux=cacher
vertical	Booléen	→ Vrai=montrer, Faux=cacher

La commande MONTRER GRILLE LISTBOX permet d'afficher ou de masquer les traits horizontaux et/ou verticaux composant la grille de l'objet list box désigné par les paramètres *objet* et *\**.

Si vous passez le paramètre optionnel *\**, vous indiquez que le paramètre *objet* est un nom d'objet (une chaîne). Si vous ne passez pas ce paramètre, vous indiquez que le paramètre *objet* est une variable. Dans ce cas, vous ne passez pas une chaîne mais une référence de variable.

Passez dans les paramètres *horizontal* et *vertical* des valeurs booléennes indiquant si les traits correspondants doivent être affichés (Vrai) ou cachés (Faux). Par défaut, la grille est affichée.

**Référence :** [FIXER COULEUR GRILLE LISTBOX](#)

## FIXER COULEUR GRILLE LISTBOX

FIXER COULEUR GRILLE LISTBOX({\*; }objet; couleur; horizontal; vertical)

Paramètres	Type	Description
*		→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
couleur	Numérique	→ Valeur de couleur RVB
horizontal	Booléen	→ Utiliser couleur pour les traits horizontaux
vertical	Booléen	→ Utiliser couleur pour les traits verticaux

La commande `FIXER COULEUR GRILLE LISTBOX` permet de modifier la couleur de la grille de l'objet list box désigné par les paramètres *objet* et *\**.

Si vous passez le paramètre optionnel *\**, vous indiquez que le paramètre *objet* est un nom d'objet (une chaîne). Si vous ne passez pas ce paramètre, vous indiquez que le paramètre *objet* est une variable. Dans ce cas, vous ne passez pas une chaîne mais une référence de variable.

Passez dans le paramètre *couleur* une valeur de couleur RVB. Pour plus d'informations sur les couleurs RVB, reportez-vous à la description de la commande `FIXER COULEURS RVB`.

Les paramètres *horizontal* et *vertical* vous permettent de spécifier les traits auxquels la couleur doit être appliquée :

- si vous passez `Vrai` dans *horizontal*, la couleur sera appliquée aux traits horizontaux de la grille. Si vous passez `Faux`, leur couleur ne sera pas modifiée.
- si vous passez `Vrai` dans *vertical*, la couleur sera appliquée aux traits verticaux de la grille. Si vous passez `Faux`, leur couleur ne sera pas modifiée.

**Référence :** [MONTRER GRILLE LISTBOX](#)

## MONTRER BARRES DEFILEMENT LISTBOX

MONTRER BARRES DEFILEMENT LISTBOX ({\*; }objet; horizontal; vertical)

Paramètres	Type	Description
*		→ Si spécifié, objet est un nom d'objet (chaîne) Si omis, objet est une variable
objet	Objet de formulaire	→ Nom d'objet (si * est spécifié) ou Variable (si * est omis)
horizontal	Booléen	→ Vrai=montrer, Faux=cacher
vertical	Booléen	→ Vrai=montrer, Faux=cacher

La commande MONTRER BARRES DEFILEMENT LISTBOX permet d'afficher ou de masquer les barres de défilement horizontale et/ou verticale dans l'objet list box désigné par les paramètres *objet* et *\**.

Si vous passez le paramètre optionnel *\**, vous indiquez que le paramètre *objet* est un nom d'objet (une chaîne). Si vous ne passez pas ce paramètre, vous indiquez que le paramètre *objet* est une variable. Dans ce cas, vous ne passez pas une chaîne mais une référence de variable.

Passez dans les paramètres *horizontal* et *vertical* des valeurs booléennes indiquant si les barres de défilement correspondantes doivent être affichées (Vrai) ou cachées (Faux). Par défaut, les barres de défilement sont affichées.

## Environnement 4D

### OUVRIR PREFERENCES 4D

OUVRIR PREFERENCES 4D (sélecteur)

Paramètres	Type	Description
sélecteur	Chaîne	→ Clé désignant un thème ou une page ou un groupe de paramètres de la boîte de dialogue des Préférences

La commande OUVRIR PREFERENCES 4D provoque l'ouverture de la boîte de dialogue des Préférences de l'application 4<sup>e</sup> Dimension courante et l'affichage du thème ou de la page correspondant à la clé passée dans le paramètre *sélecteur*.

Le paramètre *sélecteur* doit contenir une ou plusieurs "clés" indiquant un thème, une page ou un groupe de paramètres de la boîte de

dialogue des Préférences. La liste des clés utilisables est fournie [à la fin de cette section](#).

Vous pouvez passer soit un “chemin d’accès” absolu, soit un nom d’élément simple :

- **chemin d’accès absolu** : le paramètre *sélecteur* est construit de la manière suivante : /Thème{/Page{/Groupe de paramètre}}. La chaîne doit débuter par le caractère / et chaque niveau doit être séparé par un /. Par exemple, pour désigner la page Compilateur du thème Mode Structure, *sélecteur* doit contenir “/Design Mode/Compiler”.
- **nom (chemin relatif)** : dans ce cas, le paramètre *sélecteur* ne doit pas débuter par le caractère /. Il suffit de passer le nom de l’élément souhaité et 4<sup>e</sup> Dimension ouvrira le premier élément correspondant dans l’ordre de recherche suivant : groupe de paramètres -> page -> thème. Par exemple, si vous passez “Progress indicator” dans *sélecteur*, 4<sup>e</sup> Dimension ouvrira la page Options du thème Application.

Pour ouvrir directement la boîte de dialogue sur la première page, passez simplement “/” dans *sélecteur*.

La commande ouvre la page des Préférences sur l’élément spécifié dans *sélecteur*, mais tous les autres thèmes et pages restent accessibles. Il est du ressort du développeur de s’assurer que l’accès des utilisateurs aux Préférences ne risque pas de perturber le fonctionnement de l’application. Pour contrôler les actions réalisables par les utilisateurs, il est recommandé d’activer le système de gestion des droits d’accès.

Si l’élément demandé est trouvé et correctement ouvert, la variable système *OK* retourne 1 ; sinon, elle retourne 0.

- ▼ Ouverture des Préférences sur la première page :  
**OUVRIR PREFERENCES 4D("/")**
- ▼ Ouverture de la page “Raccourcis clavier” du thème “Application” :  
**OUVRIR PREFERENCES 4D("/Application/Shortcuts")**
- ▼ Ouverture de la page “Editeur de méthodes” du thème “Mode Structure” :  
**OUVRIR PREFERENCES 4D("Default Display")**

### Clés de chemin

Voici la liste des clés utilisables dans le paramètre *sélecteur* :

- /Application
- /Application/Options
- /Application/Options/Options
- /Application/Options/Temporary Folder Location
- /Application/Options/Drag and Drop Highlight
- /Application/Options/Progress Indicator
- /Application/Options/Display Toolbar
- /Application/Options/Display Windows
- /Application/Access
- /Application/Access/Data Access
- /Application/Access/User Access
- /Application/CPU Priorities
- /Application/CPU Priorities/Set CPU Priority to:
- /Application/Shortcuts
- /Application/Shortcuts/Keys
- /Application/Compatibility
- /Application/Compatibility/Structure Compatibility
- /Application/Compatibility/Web Compatibility
- /Application/Compatibility/Platform
- /Design Mode
- /Design Mode/Structure
- /Design Mode/Structure/General Font
- /Design Mode/Structure/Forms and Methods Automatic Comments
- /Design Mode/Form Editor
- /Design Mode/Form Editor/Object Templates
- /Design Mode/Form Editor/Move
- /Design Mode/Form Editor/Auto Alignment
- /Design Mode/Form Editor/Default Display
- /Design Mode/Method Editor
- /Design Mode/Method Editor/Font
- /Design Mode/Method Editor/Default Display
- /Design Mode/Method Editor/Options
- /Design Mode/Method Editor/Structure Style-Sheets
- /Design Mode/Compiler

/Design Mode/Compiler/Compilation Options  
/Design Mode/Compiler/Compiler Methods for...  
/Design Mode/Documentation  
/Design Mode/Documentation/Documentation Access from the Explorer  
/Database  
/Database/Data Management  
/Database/Data Management/General  
/Database/Data Management/Database Cache Settings  
/Database/Data Management/WEDD  
/Database/Script Manager  
/Database/Script Manager/Script Manager  
/Backup  
/Backup/Configuration  
/Backup/Configuration/Backup Contents  
/Backup/Configuration/Backup File Destination Folder  
/Backup/Configuration/Last Backup Information  
/Backup/Configuration/Log Management  
/Backup/Scheduler  
/Backup/Scheduler/Backup Frequency  
/Backup/Backup  
/Backup/Backup/General  
/Backup/Backup/Archive  
/Backup/Restore  
/Backup/Restore/Automatic Restore  
/Client-Server  
/Client-Server/Configuration  
/Client-Server/Configuration/Network  
/Client-Server/Configuration/Client-Server Connections Timeout  
/Client-Server/Configuration/Client-Server Communication  
/Client-Server/Configuration/4D Open  
/Client-Server/Publishing  
/Client-Server/Publishing/Publishing Information  
/Client-Server/Publishing/Allow-Deny Table Configuration  
/Client-Server/Publishing/Encryption  
/Web  
/Web/Configuration

/Web/Configuration/Web Server Publishing  
/Web/Configuration/Default HTML Path  
/Web/Configuration/Starting Mode  
/Web/Advanced  
/Web/Advanced/Cache  
/Web/Advanced/Web Process  
/Web/Advanced/Options  
/Web/Advanced/Web Passwords  
/Web/Options  
/Web/Options/Text Conversion  
/Web/Options/4D WebSTAR  
/Web/Options/Persistent Connections  
/Web Services  
/Web Services/SOAP  
/Web Services/SOAP/Server Side  
/Web Services/SOAP/Client Side

## Environnement système

### FIXER VARIABLE ENVIRONNEMENT

FIXER VARIABLE ENVIRONNEMENT(*nomVar*; *valeurVar*)

Paramètres	Type	Description
<i>nomVar</i>	Chaîne	→ Nom de la variable à fixer
<i>valeurVar</i>	Chaîne	→ Valeur de la variable ou "" pour rétablir la valeur par défaut

Cette commande vous permet de fixer la valeur d'une variable d'environnement sous MacOS X et Windows. Elle est destinée à une utilisation conjointe avec la commande [FIXER EXECUTABLE CGI](#) ou LANCER PROCESS EXTERNE.

Passez dans le paramètre *nomVar* le nom de la variable à définir et dans le paramètre *valeurVar* sa valeur.

Pour obtenir la liste générale des variables d'environnement et leurs valeurs possibles, reportez-vous à la documentation technique de votre système d'exploitation.

Pour connaître la liste des variables d'environnement utilisables avec la commande [FIXER EXECUTABLE CGI](#), reportez-vous à la documentation



4D concernant les CGI.

Pour connaître la liste des variables d'environnement utilisables avec la commande LANCER PROCESS EXTERNE, reportez-vous à la documentation de cette commande.

## LANCER PROCESS EXTERNE

LANCER PROCESS EXTERNE(nomFichier{; fluxEntrée{; fluxSortie{; fluxErreur}}})

Paramètres	Type	Description
nomFichier	Chaîne	→ Chemin d'accès et arguments du fichier à lancer
fluxEntrée	Chaîne	→ Flux d'entrée (stdin)
fluxSortie	Chaîne	← Flux de sortie (stdout)
fluxErreur	Chaîne	← Flux d'erreur (stderr)

*Note* Cette commande est destinée aux utilisateurs avancés.

La commande LANCER PROCESS EXTERNE permet de lancer un process externe depuis 4<sup>e</sup> Dimension, sous MacOS X et Windows.

Sous MacOS X, cette commande donne accès à toutes les applications pouvant être lancées depuis le Terminal.

*Note* Pour les utilisateurs de 4D Pack, cette commande reprend et étend les possibilités offertes par la commande AP\_Sublaunch.

Passez dans le paramètre *nomFichier* le chemin d'accès absolu de l'application à exécuter ainsi que les arguments nécessaires, le cas échéant.

Sous MacOS X, vous pouvez également passer uniquement le nom de l'application à exécuter, 4<sup>e</sup> Dimension utilisera alors la variable d'environnement *PATH* pour localiser l'exécutable.

Le paramètre *fluxEntrée* (facultatif) contient le stdin du process externe. Après l'exécution de la commande, les paramètres *fluxSortie* et *fluxErreur* (s'ils sont passés) retournent respectivement le stdout et le stderr du process externe.

Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1. Sinon (fichier non trouvé, mémoire insuffisante, etc.), elle prend la valeur 0.

**Exemples sous MacOS X** Tous les exemples suivants utilisent le Terminal de MacOS X accessible dans le dossier Applications/Utilitaires.

- ▼ Pour modifier les accès à un fichier (chmod est la commande MacOS X permettant de modifier les accès des fichiers) :

```
LANCER PROCESS EXTERNE ("chmod +x /dossier/monfichier.txt")
```

- ▼ Pour éditer un fichier texte (cat est la commande MacOS X permettant d'éditer les fichiers). Dans cet exemple, le chemin d'accès absolu de la commande est passé :

```
C_TEXTE(entrée;sortie)  
entrée:=""
```

```
LANCER PROCESS EXTERNE ("/bin/cat /dossier/monfichier.txt";  
entrée;sortie)
```

- ▼ Pour récupérer le contenu du dossier "Users" (ls -l est semblable à la commande dir du DOS) :

```
C_TEXTE($in;$out)  
LANCER PROCESS EXTERNE("bin/ls -l /Users";$in;$out)
```

### Exemple sous Windows

- ▼ Pour lancer l'application NotePad :

```
LANCER PROCESS EXTERNE ("C:\\WINDOWS\\notepad.exe")
```

- ▼ Pour lancer l'application NotePad et ouvrir un document spécifique :

```
LANCER PROCESS EXTERNE ("C:\\WINDOWS\\notepad.exe C:\\Docs  
\\nouveau dossier\\res.txt")
```

- ▼ Pour lancer l'application Microsoft® Word® et ouvrir un document spécifique (à noter l'emploi de deux "" ) :

```
$mondoc:="C:\\Program Files\\Microsoft Office\\Office10  
\\WINWORD.EXE \"C:\\Documents and Settings\\  
JeanMarc\\Bureau\\MesDocs\\  
Nouveau dossier\\essai.xml\""  
LANCER PROCESS EXTERNE($mondoc;$tIn;$tOut)
```

- ▼ Pour exécuter un script Perl (requiert l'installation préalable d'ActivePerl) :

```
C_TEXTE($entrée;$sortie)  
FIXER VARIABLE ENVIRONNEMENT("mavARIABLE";"valeur")  
LANCER PROCESS EXTERNE ("D:\\Perl\\bin\\perl.exe D:\\Perl\\eg  
\\cgi\\env.pl";$entrée;$sortie)
```

## Sélections

De nouvelles possibilités de gestion des sélections dans les formulaires listes et les sous-formulaires ont été ajoutées dans 4<sup>e</sup> Dimension 2004. Ces nouveautés se traduisent par :

- la création des commandes **LIRE ENREGISTREMENTS MARQUES** et **Numero de ligne affichee**,
- la modification de la commande **MARQUER ENREGISTREMENTS**,
- la modification des commandes **VISUALISER SELECTION** et **MODIFIER SELECTION**.

---

*Note* D'autres nouveautés contribuent également à faciliter la gestion des formulaires listes et des sous-formulaires :

- la commande **CHOIX SAISSABLE** (thème "Propriétés des objets") permet désormais de fixer le mode Saisie en liste des formulaires liste,
- le nouvel événement formulaire **Sur changement sélection** est généré dans les listes et les sous-formulaires (cf. [paragraphe "Événements formulaires"](#), page 301),
- la nouvelle commande **EDITER ELEMENT** (thème "Propriétés des objets") permet de passer une ligne de sous-formulaire en édition.

---

### LIRE ENREGISTREMENTS MARQUES

LIRE ENREGISTREMENTS MARQUES({table; }nomEnsemble)

Paramètres	Type	Description
table	table	→ Table de laquelle recopier l'ensemble UserSet Si omis, table du formulaire courant
nomEnsemble	Chaîne	→ Ensemble dans lequel recopier l'ensemble UserSet

La commande **LIRE ENREGISTREMENTS MARQUES** recopie dans l'ensemble désigné par le paramètre *nomEnsemble* l'ensemble UserSet courant de la *table* passée en paramètre. Si le paramètre *table* est omis, la table du formulaire ou du sous-formulaire courant est utilisée.

**Rappel** : L'ensemble système UserSet d'une table contient l'ensemble des enregistrements "marqués" par l'utilisateur dans un formulaire sortie. Il existe un seul ensemble UserSet par table et par process.

Comme elle permet de désigner la table de laquelle récupérer les enregistrements marqués, la commande **LIRE ENREGISTREMENTS**

MARQUES peut notamment gérer les sélections d'enregistrements dans les sous-formulaires inclus. En effet dans ce cas, les sélections des sous-formulaires peuvent provenir de tables différentes. Dans les versions précédentes de 4<sup>e</sup> Dimension, la gestion des sélections effectuées par l'utilisateur s'effectuait uniquement via l'ensemble UserSet et était donc limitée à la table courante.

La commande LIRE ENREGISTREMENTS MARQUES peut être appelée hors du contexte d'un formulaire, cependant dans ce cas l'ensemble retourné est vide.

L'ensemble désigné par le paramètre *nomEnsemble* peut être local/client, process ou interprocess.

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

- ▼ Cette méthode indique combien d'enregistrements sont sélectionnés dans le sous-formulaire affichant les enregistrements de la table [CDs] :

```
LIRE ENREGISTREMENTS MARQUES ([CDs];"$highlight")
ALERTE(Chaine(Enregistrements dans ensemble("$highlight"))+
      " enregistrements sélectionnés.")
EFFACER ENSEMBLE("$highlight")
```

*Référence* : [MARQUER ENREGISTREMENTS](#)

## Numero de ligne affichee

Numero de ligne affichee → Entier long

Paramètres	Type	Description
		Cette commande ne requiert pas de paramètre

Résultat	Entier long ←	Numéro de ligne en cours d'affichage
----------	---------------	--------------------------------------

La commande Numero de ligne affichee fonctionne uniquement dans le contexte de l'événement formulaire Sur affichage corps. Elle retourne le numéro de la ligne en cours de traitement durant l'affichage à l'écran d'une liste d'enregistrements.

Lorsque la ligne affichée n'est pas vide (c'est-à-dire lorsqu'elle est associée à un enregistrement), la valeur retournée par Numero de ligne affichee est identique à celle retournée par Numero dans selection.

Comme Numero dans selection, Numero de ligne affichee débute à 1.

Cette commande est utile lorsque vous souhaitez appliquer un traitement à chaque ligne d'un formulaire liste affiché à l'écran, y compris aux lignes vides.

Si Numero de ligne affichee est appelée en-dehors de l'affichage d'une liste, elle retourne 0.

- ▼ Cet exemple permet d'appliquer une couleur alternée à un formulaire liste affiché à l'écran, même pour les lignes sans enregistrement :

`Méthode du formulaire liste

**Si (Evenement formulaire=Sur affichage corps)**

**Si (Numero de ligne affichee % 2 = 0)**

    `Noir sur blanc pour le texte des lignes paires

**FIXER COULEURS RVB( [Table 1]Champ1 ; 0 ; 0x00FFFFFF);**

**Sinon**

    `Noir sur bleu pâle pour le texte des lignes impaires

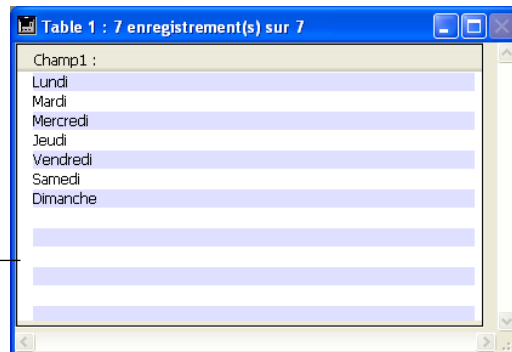
**FIXER COULEURS RVB( [Table 1]Champ1 ; 0 ; 0x00E0E0FF)**

**Fin de si**

**Fin de si**

Lignes associées à  
des enregistrements

Lignes vides



## MARQUER ENREGISTREMENTS

MARQUER ENREGISTREMENTS ({{table; }nomEnsemble}{;\* })

Paramètres	Type	Description
table	table	→ Table de laquelle marquer les enregistrements Si omis, table du formulaire courant
nomEnsemble	Chaîne	→ Ensemble d'enregistrements à marquer ou Ensemble UserSet si ce paramètre est omis
*		→ Inactiver le défilement automatique de la liste

La commande MARQUER ENREGISTREMENTS admet deux paramètres supplémentaires.

Le paramètre *table* permet de désigner la table de laquelle les enregistrements doivent être “marqués”. Ce nouveau paramètre permet notamment de marquer les enregistrements des sous-formulaires inclus — n’appartenant donc pas à la table courante.

Le paramètre \*, s’il est passé, provoque l’inactivation de la fonction de défilement automatique de la liste si les enregistrements marqués ne sont pas visibles. Ce mécanisme autorise la gestion personnalisée du défilement via la nouvelle commande DEFILER LIGNES.

**Référence :** LIRE ENREGISTREMENTS MARQUES, DEFILER LIGNES

**MODIFIER SELECTION, VISUALISER SELECTION**

MODIFIER SELECTION({table} {; modeSélection}{; saisieListe} {; \*{; \*}})  
 VISUALISER SELECTION({table} {; modeSélection}{; saisieListe} {; \*{; \*}})

Paramètres	Type	Description
table	Table	→ Table à afficher et modifier, ou Table par défaut si ce paramètre est omis
modeSélection	Entier long	→ Mode de sélection
saisieListe	Booléen	→ Autoriser saisie en liste
*	*	→ Utiliser formulaire sortie pour un seul enregistrement et cacher les barres de défilement dans le formulaire entrée
*	*	→ Afficher les barres de défilement dans le formulaire entrée (= annuler le second effet du premier paramètre *)

Les deux commandes MODIFIER SELECTION et VISUALISER SELECTION admettent deux paramètres facultatifs supplémentaires, permettant de définir les options de sélection et de saisie en liste des formulaires affichés.

Par ailleurs, ces commandes n’affichent plus par défaut le bouton **Quitter la liste**.

Le paramètre *modeSélection* vous permet de définir les possibilités de sélection d’enregistrements dans la liste. Vous pouvez passer dans ce paramètre une des constantes du thème “Options de formulaire” :

- si vous passez Pas de sélection (0), il ne sera pas possible de sélectionner un enregistrement dans la liste.

- si vous passez Sélection unique (1), seule la sélection d'un enregistrement à la fois sera autorisée.
- si vous passez Sélection multiple (2), l'utilisateur pourra sélectionner plusieurs enregistrements simultanément, à l'aide des touches **Maj** et/ou **Ctrl** (Windows) ou **Commande** (MacOS).

Par défaut, si vous ne passez pas le paramètre *modeSélection*, le mode "Sélection multiple" est utilisé.

Le paramètre *saisieListe* vous permet d'autoriser le mode "Saisie en liste" dans la liste affichée. Passez **Vrai** pour autoriser ce mode ou **Faux** pour ne pas l'autoriser.

Par défaut, si vous ne passez pas le paramètre *saisieListe*, le mode "Saisie en liste" n'est pas autorisé.

---

*Note* La commande **CHOIX SAISSISSABLE** permet désormais d'activer ou de désactiver le mode Saisie en liste à la volée (cf. [commande CHOIX SAISSISSABLE](#), page 323).

---

## Quitter la liste

4<sup>e</sup> Dimension 2004 n'affiche plus le bouton par défaut **Quitter la liste** lorsqu'un formulaire utilisé par la commande MODIFIER SELECTION ou VISUALISER SELECTION est dépourvu de boutons. Dans ce cas désormais, seule la touche **Echap** (Windows) ou **Esc** (MacOS) permet de quitter la liste.

## ALLER DANS SELECTION

ALLER DANS SELECTION ({table; }position)

Paramètres	Type	Description
table	Table	→ Table dans laquelle aller à l'enregistrement spécifié ou Table par défaut si omis
position	Numérique	→ Position de l'enregistrement dans la sélection

La commande ALLER DANS SELECTION admet désormais la valeur 0 dans le paramètre *position*. Lorsque cette valeur est passée, il n'y a plus d'enregistrement courant dans *table*. Ce nouveau mécanisme permet de désélectionner tous les enregistrements d'une liste, notamment dans les sous-formulaires inclus.

## Serveur Web

Le thème “Serveur Web” du langage de 4<sup>e</sup> Dimension 2004 s’est enrichi de plusieurs nouvelles commandes : [TRAITER BALISES HTML](#), [LIRE CORPS HTTP](#) et [ENVOYER DONNEES HTTP](#). En outre, les capacités de la commande [LIRE VARIABLES FORMULAIRE WEB](#) ont été étendues.

A noter également l’ajout d’une commande permettant une meilleure intégration des CGI : [FIXER EXECUTABLE CGI](#).

---

*Note* La nouvelle commande [ENVOYER DONNEES HTTP](#) s’inscrit parmi l’ensemble de nouveautés liées à la prise en charge de HTTP/1.1 par le serveur Web de 4<sup>e</sup> Dimension. Pour plus d’informations sur ce point, reportez-vous au [chapitre “Serveur Web”, page 241](#).

---

### TRAITER BALISES HTML

TRAITER BALISES HTML(donnéesEntrée; donnéesSortie)

Paramètres	Type	Description
donnéesEntrée	Texte ou BLOB →	Données contenant des balises HTML à traiter
donnéesSortie	Texte ou BLOB ←	Données traitées

La commande `TRAITER BALISES HTML` provoque le traitement par 4<sup>e</sup> Dimension des balises HTML 4D contenues dans le paramètre *donnéesEntrée* (champ ou variable de type Texte ou BLOB) et retourne les données résultantes dans *donnéesSortie*.

Cette commande permet d’effectuer un traitement sur du code HTML balisé sans qu’il soit nécessaire que le serveur Web envoie une page HTML via une commande du type `ENVOYER BLOB HTML` ou qu’une page suffixée “.shtml” soit demandée via un URL. Il n’est même pas nécessaire que le serveur Web de 4<sup>e</sup> Dimension soit démarré.

Passez les données contenant les balises à traiter dans le paramètre *donnéesEntrée*. Ce paramètre peut être un champ ou une variable de type Texte ou BLOB. A noter qu’en général, le type BLOB est préférable car il n’est pas limité en nombre de caractères (32000 pour le type Texte).

Toutes les balises HTML de 4D sont prises en charge (4DVAR, 4DSCRIPT, 4DLOOP, etc.), quel que soit le mode de fonctionnement du serveur Web (contextuel ou sans contexte) — et même s’il n’est pas lancé. Pour



plus d'informations sur ces balises, reportez-vous à la section "Balises HTML 4D" dans le manuel *Langage de 4<sup>e</sup> Dimension*.

*Note* En cas d'utilisation de la balise 4DINCLUDE hors du cadre du serveur Web (process Web) :

- avec 4<sup>e</sup> Dimension et 4D Server, le dossier par défaut est le dossier contenant le fichier de structure de la base,
- avec 4D Client, le dossier par défaut est le dossier contenant l'application 4D Client.

Après l'exécution de la commande, le paramètre *donnéesSortie* reçoit les données du paramètre *donnéesEntrée* ainsi que le résultat du traitement des balises HTML 4D qu'il contenait, le cas échéant. Si *donnéesEntrée* ne contenait pas de balises HTML 4D, le contenu de *donnéesSortie* est identique à celui de *donnéesEntrée*.

Le paramètre *donnéesSortie* peut être un champ ou une variable, il doit simplement être du même type que le paramètre *donnéesEntrée*.

Cette commande rend possible le stockage dans la base de valeurs issues d'un traitement de balises HTML *avant* qu'elles ne soient envoyées.

Elle permet également d'analyser des balises HTML 4D en-dehors de l'utilisation du serveur Web. En particulier, vous pouvez l'employer pour envoyer via les 4D Internet Commands des courriels au format HTML contenant des traitements et/ou des références à des données contenues dans la base.

- ▼ Cet exemple illustre le fonctionnement de cette commande :

```
C_BLOB($in)
C_BLOB($out)
C_TEXTE($in_text)
C_TEXTE(Var)
C_TEXTE(VarHTML)
```

```
Var:="<B>"
$in_text:="<p><!--#4DVAR Var--></p>"
TEXTE VERS BLOB($in_text;$in;Texte sans longueur )
TRAITER BALISES HTML($in;$out)
VarHTML:=BLOB vers texte($out;Texte sans longueur )
  ` VarHTML contient <p>&lt;B&gt;</p>
```

## LIRE CORPS HTTP

### LIRE CORPS HTTP(corps)

Paramètres	Type	Description
corps	BLOB ou Texte	← Champ corps (Body) de la requête HTTP

La commande LIRE CORPS HTTP retourne le corps (*body*) de la requête HTTP en cours de traitement. Le corps HTTP est retourné tel quel, sans traitement ni analyse.

Cette commande fonctionne uniquement en mode sans contexte. Elle peut être appelée depuis une méthode base Web (Sur authentification Web, Sur connexion Web) ou toute méthode Web exécutée en mode sans contexte.

Vous pouvez passer dans le paramètre *corps* une variable ou un champ de type BLOB ou Texte. A noter qu'en général, le type BLOB est préférable car il n'est pas limité en nombre de caractères. Le type Texte, quant à lui, est limité à 32 000 caractères ; en cas de dépassement, les données reçues seront tronquées.

Cette commande permet par exemple d'effectuer des recherches dans le corps des requêtes. Elle permet également aux utilisateurs avancés de mettre en place un serveur WebDAV au sein d'une base 4<sup>e</sup> Dimension.

- ▼ Dans cet exemple, une requête simple est envoyée au serveur Web de 4<sup>e</sup> Dimension et le contenu du champ HTTP *corps* est visualisé dans le débogueur.

Voici le formulaire envoyé au serveur Web de 4<sup>e</sup> Dimension, ainsi que le code HTML correspondant :

Formulaire

```

TEST

Nom


    
```

Corps

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <title>Page de test</title>
  </head>
  <body>
    TEST
    <FORM ACTION="/4DACTION/Test4D2004" METHOD=POST>
      <br>Nom<br>
      <input type="text" value="Saisissez votre nom" name="vnom">
      <br>
      <input type="submit">
    </FORM>
  </body>
</html>
    
```

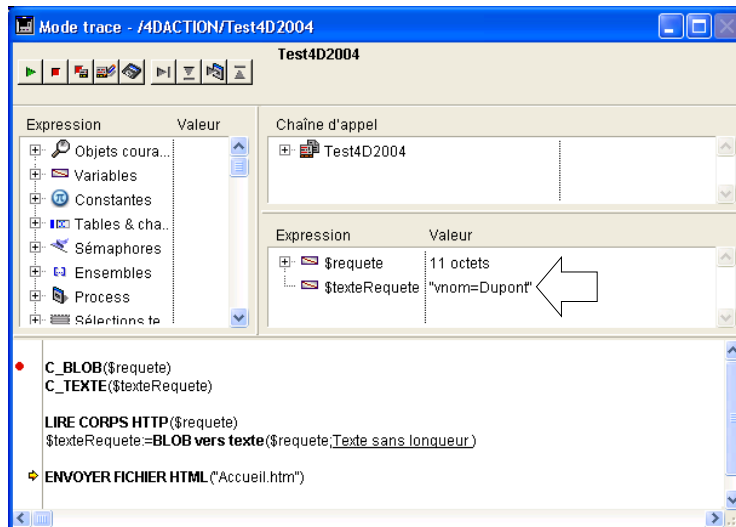
Voici la méthode *Test4D2004* :

**C\_BLOB(\$requete)**  
**C\_TEXTE(\$texteRequete)**

**LIRE CORPS HTTP(\$requete)**  
**\$texteRequete:=BLOB vers texte(\$requete;Texte sans longueur)**  
**ENVOYER FICHER HTML("page.html")**

*Note* Cette méthode a été déclarée "Disponible via 4DACTION..." dans ses propriétés.

Lorsque le formulaire est soumis au serveur Web, la variable *\$texteRequete* reçoit le texte du champ corps de la requête HTTP :



## ENVOYER DONNEES HTTP

ENVOYER DONNEES HTTP(données{; \*})

Paramètres	Type	Description
données	BLOB	→ Données HTTP à envoyer
*	*	→ Envoi morcelé (chunked)

Cette commande permet au serveur Web 4D d'envoyer des données HTTP "brutes", pouvant être morcelées. Elle fonctionne uniquement en mode sans contexte.

Le paramètre *données* contient les deux parties standard d'une réponse HTTP, c'est-à-dire l'en-tête et le corps (*header* et *body*). Les données sont envoyées sans formatage préalable par le serveur.

Toutefois, 4<sup>e</sup> Dimension effectue un contrôle élémentaire sur l'en-tête et le corps de la réponse afin qu'elle soit valide :

- Si l'en-tête est incomplet ou non conforme aux spécifications du protocole HTTP, 4<sup>e</sup> Dimension le modifie en conséquence.
- Si la réponse HTTP est incomplète, 4<sup>e</sup> Dimension ajoute les informations manquantes. Si, par exemple, vous souhaitez effectuer une redirection, vous devez écrire :

HTTP/1.1 302

Location : http://...

Si vous passez uniquement :

Location : http://...

4<sup>e</sup> Dimension complétera la réponse en ajoutant HTTP/1.1 302.

Le paramètre optionnel \* permet de déclarer que la réponse sera envoyée sous forme "morcelée" (*chunked*). Le découpage des réponses peut être utile lorsque le serveur envoie une réponse sans connaître sa longueur totale (par exemple si la réponse n'a pas encore été générée). Tous les navigateurs compatibles HTTP/1.1 acceptent les réponses "morcelées".

Si vous passez le paramètre \*, le serveur Web inclura automatiquement le champ **transfer-encoding: chunked** dans l'en-tête de la réponse, si nécessaire (vous pouvez gérer manuellement l'en-tête de la réponse si vous le souhaitez). Le reste de la réponse sera également formaté en respectant la syntaxe de l'option *chunked*. Les réponses morcelées comportent un seul en-tête et un nombre indéfini de corps.

Toutes les instructions ENVOYER DONNEES HTTP suivant l'exécution de ENVOYER DONNEES HTTP(données;\*) au sein de la même méthode seront considérées comme partie de la réponse (qu'elles contiennent ou non le paramètre \*). Le serveur met un terme à l'envoi morcelé à la fin de l'exécution de la méthode.

---

*Note* Si le client Web ne prend pas en charge le protocole HTTP/1.1, 4<sup>e</sup> Dimension convertira automatiquement la réponse au format compatible HTTP/1.0 (l'envoi ne sera pas morcelé). Dans ce cas toutefois, il est possible que le résultat ne corresponde pas à vos attentes. Il est donc recommandé de tester si le navigateur est compatible HTTP/1.1 et d'envoyer une réponse adaptée. Pour cela, vous pouvez utiliser une méthode du type ci-dessous :

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

**TABLEAU TEXTE(tabChamps;0)**

**TABLEAU TEXTE(tabValeurs;0)**

**LIRE ENTETE HTTP**(tabChamps;tabValeurs)

\$0:=Faux

Si (Taille tableau(tabValeurs)>=3)

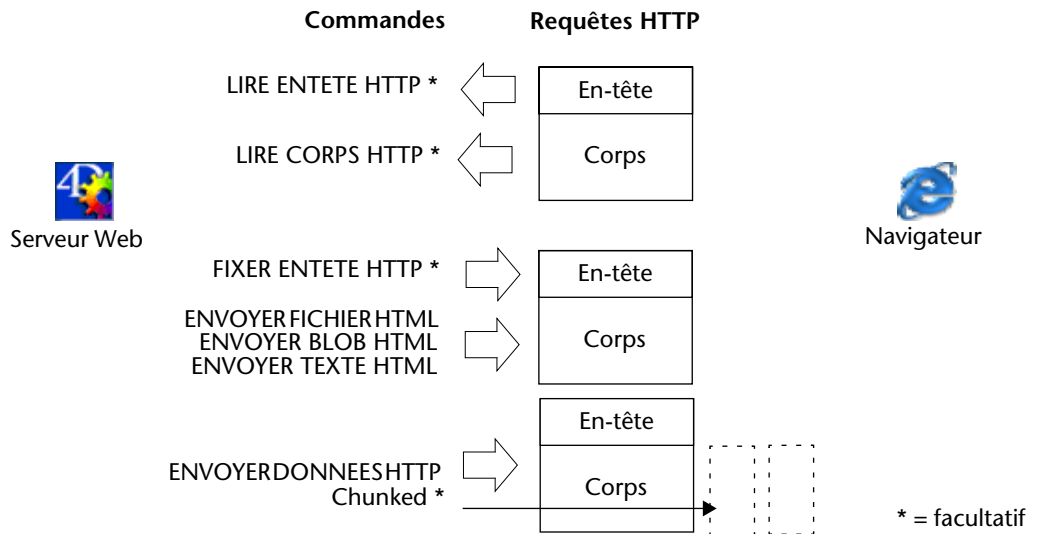
Si (Position("HTTP/1.1";tabValeurs{3})>0)

\$0:=Vrai ` Le navigateur est compatible HTTP/1.1

Fin de si

Fin de si

Combinée à la nouvelle commande **LIRE CORPS HTTP** et aux autres commandes du thème “Serveur Web”, cette commande complète la gamme d’outils mis à la disposition des développeurs 4D pour traiter de manière entièrement personnalisée les connexions HTTP entrantes et sortantes. Ces différents outils sont présentés dans le schéma suivant :



- ▼ Cet exemple illustre l’emploi de l’option *chunked* avec la commande ENVOYER DONNEES HTTP. Les données (une suite de chiffres) sont envoyées en 100 morceaux générés à la volée dans une boucle. A noter que l’en-tête de la réponse n’est pas explicitement défini : la commande ENVOYER DONNEES HTTP l’enverra automatiquement et y insérera le champ transfer-encoding: chunked car le paramètre \* est utilisé.

C\_ENTIER LONG(\$cpt)

C\_BLOB(\$mon\_blob)

C\_TEXTE(\$output)

```
Boucle ($cpt;1;100)
  $output:="[ "+Chaine($cpt)+"]"
  TEXTE VERS BLOB($output;$mon_blob;Texte sans longueur)
  ENVOYER DONNEES HTTP($mon_blob;*)
Fin de boucle
```

## LIRE VARIABLES FORMULAIRE WEB

Les capacités de cette commande ont été étendues dans 4<sup>e</sup> Dimension 2004 : il est désormais possible de l'utiliser dans tous les cas, quel que soit le type d'URL envoyé au serveur Web. En particulier, elle fonctionne maintenant avec les formulaires HTML envoyant des données POST vers tout URL.

Dans les versions précédentes de 4<sup>e</sup> Dimension, seules les requêtes débutant par /4DACTION, /4DMETHOD et /4DCGI ou contenant une chaîne d'interrogation pouvaient être analysées par cette commande.

## Prise en charge des CGI

La prise en charge des CGI a été simplifiée et étendue dans 4<sup>e</sup> Dimension 2004. Le serveur Web de 4<sup>e</sup> Dimension peut désormais faire appel à tout type de CGI, sous MacOS X et sous Windows.

---

*Note* L'ancien mécanisme de communication CGI entre 4<sup>e</sup> Dimension et 4D WebSTAR basé sur les Apple Events n'est plus pris en charge dans 4<sup>e</sup> Dimension 2004.

---

La modification de la prise en charge des CGI dans 4<sup>e</sup> Dimension 2004, se traduit par l'ajout de deux nouvelles commandes :

- **FIXER EXECUTABLE CGI**,
- **FIXER VARIABLE ENVIRONNEMENT** (thème "Environnement système"). Cette commande est placée dans le thème "Environnement système" car sa portée ne se limite pas au serveur Web de 4<sup>e</sup> Dimension.

## FIXER EXECUTABLE CGI

FIXER EXECUTABLE CGI (url1 {; url2})

Paramètres	Type	Description
url1	Chaîne	→ URL d'accès
url2	Chaîne	→ URL d'accès

La commande **FIXER EXECUTABLE CGI** permet d'exécuter un CGI sans que celui-ci soit visible pour l'utilisateur Web dans l'URL.

Cette commande peut notamment être utilisée dans la méthode base Sur authentification Web, afin de déterminer par exemple le CGI à exécuter. Elle fonctionne sous MacOS X et sous Windows.

Passez dans le paramètre *url1* l'URL d'accès au CGI à exécuter. Par exemple, si vous écrivez `FIXER EXECUTABLE CGI("/myfile.pl")`, le serveur Web 4D exécutera le CGI `myfile.pl` — cette application doit se trouver dans le dossier par défaut du serveur Web.

Si vous passez une chaîne vide ("" ) dans *url1*, 4<sup>e</sup> Dimension exécutera directement le CGI défini dans l'URL envoyé par le navigateur, le cas échéant.

Passez dans le paramètre facultatif *url2* l'URL d'accès à un fichier devant être traité par le CGI. Par exemple, si vous écrivez `FIXER EXECUTABLE CGI("cgi-bin/Perl2.cgi";"Perl2.pl")`, le serveur Web exécutera le CGI `Perl2.cgi` (situé dans le dossier `cgi-bin`) en lui passant le fichier `Perl2.pl`.

Si vous passez une chaîne vide ("" ) dans *url2*, 4<sup>e</sup> Dimension passera au CGI pour traitement le fichier défini dans l'URL envoyé par le navigateur. Ce mécanisme est utilisé notamment par PHP. Exemple : `FIXER EXECUTABLE CGI("/cgi-bin/php";"")`.

Si l'URL d'accès spécifié par la commande est incorrect, le navigateur affichera la page d'erreur "Fichier non trouvé".

A noter que la commande `FIXER EXECUTABLE CGI` ne retourne pas directement d'erreur. Cette commande définit uniquement une "valeur courante" qui est utilisée ultérieurement, lorsque le CGI sera appelé. En cas d'appels multiples de cette commande, seule la valeur définie par le dernier appel sera utilisée.

- ▼ Dans cet exemple, le fichier `exemple.php`, non situé dans le dossier `cgi-bin`, est traité par le CGI `Perl2.cgi`, situé dans ce dossier :

```
FIXER EXECUTABLE CGI("/cgi-bin/Perl2.cgi";"exemple.php")
```

## Tableaux

Le thème "Tableaux" contient deux nouvelles commandes permettant de compter le nombre d'occurrences d'une valeur dans un tableau et d'ajouter un élément à la fin d'un tableau.

### Compter dans tableau

Compter dans tableau (tableau; valeur) → Entier long

Paramètres	Type	Description
tableau	Tableau	→ Tableau dans lequel effectuer le comptage
valeur	Expression	→ Valeur à compter
Résultat	Entier long	← Nombre d'occurrences trouvées

Compter dans tableau retourne le nombre d'occurrences de *valeur* dans *tableau*.

Cette commande peut être utilisée avec des tableaux de type Texte, Alpha, Numérique, Date, Pointeur et Booléen. Les paramètres *tableau* et *valeur* doivent être du même type ou d'un type compatible.

Si aucun élément de *tableau* ne correspond à *valeur*, la commande retourne 0.

- ▼ L'exemple suivant permet d'afficher le nombre de lignes sélectionnées dans une list box.

```
` tBList est le nom d'une variable d'objet List box
ALERTE(Chaine(Compter dans tableau(tBList;Vrai))+" ligne(s)
sélectionnée(s) dans la list box")
```

### AJOUTER A TABLEAU AJOUTER A TABLEAU(tableau; valeur)

Paramètres	Type	Description
tableau	Tableau	→ Tableau auquel ajouter une valeur
valeur	Expression	→ Valeur à ajouter au tableau

La commande AJOUTER A TABLEAU ajoute une nouvelle ligne à la fin du *tableau* et lui affecte la valeur passée dans le paramètre *valeur*. En mode interprété, si le *tableau* n'a pas été défini au préalable, la commande le crée et lui attribue un type en fonction de celui de *valeur*.

Cette commande fonctionne avec tous les types de tableaux : chaîne, numérique, booléen, date, pointeur et image.



Le type de *valeur* doit correspondre au type du tableau, sinon l'erreur de syntaxe 54 "Les arguments sont incompatibles" est générée. Les combinaisons suivantes sont toutefois possibles :

- un *tableau* de type chaîne (Texte ou Alpha) accepte toute *valeur* de type Texte ou Alpha.
  - un *tableau* de type numérique (Entier, Entier long ou Réel) accepte toute *valeur* de type Entier, Entier long, Numérique ou Heure.
- ▼ Le code suivant :

```
INSERER LIGNES($montableau;Taille tableau($montableau)+1)
$montableau{Taille tableau($montableau)}:=$mavaleur
```

peut désormais être remplacé par :

```
AJOUTER A TABLEAU($montableau;$mavaleur)
```

## Communications

Ce thème contient une nouvelle commande permettant d'établir la correspondance entre les numéros et les noms des ports série.

### LIRE CORRESPONDANCE PORT SERIE

```
LIRE CORRESPONDANCE PORT SERIE(tabNums; tabNoms)
```

Paramètres	Type	Description
tabNums	Tableau numérique	← Tableau de numéro(s) de port série
tabNoms	Tableau texte	← Tableau de nom(s) de port série

La commande LIRE CORRESPONDANCE PORT SERIE retourne deux tableaux *tabNums* et *tabNoms* contenant respectivement la liste des numéros et des noms des ports série de la machine courante.

Cette commande est utile sous MacOS X car le système alloue dynamiquement les numéros des ports série lorsque vous utilisez un adaptateur série USB. A l'aide de cette commande, vous pouvez adresser les ports série via leur nom (invariable), quel que soit leur numéro.

## Sauvegarde

Le thème “Sauvegarde” contient les commandes de langage associées au nouveau module de sauvegarde intégré dans 4<sup>e</sup> Dimension 2004 : SAUVEGARDER, RESTITUER, LIRE INFORMATION SAUVEGARDE et LIRE INFORMATION RESTITUTION.

En outre, deux nouvelles méthodes bases ont également été créées pour la gestion programmée des sauvegardes : Méthode base Sur démarrage sauvegarde et Méthode base Sur arrêt sauvegarde.

Pour plus d’informations sur le fonctionnement général de ce module, reportez-vous au chapitre “Module de sauvegarde intégré”, page 197.

### SAUVEGARDER

#### SAUVEGARDER

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

Cette commande ne requiert pas de paramètre

La commande SAUVEGARDER déclenche la sauvegarde de la base de données avec les paramètres de sauvegarde courants. Aucune boîte de dialogue de confirmation n’est affichée. Une fenêtre de progression apparaît à l’écran.

Les paramètres de sauvegarde sont définis dans les Préférences de l’application. Ils sont également stockés dans le fichier Backup.XML situé dans le sous-dossier Preferences/Backup de la base.

La commande SAUVEGARDER appelle les méthodes base Sur démarrage sauvegarde au début de son exécution et Sur arrêt sauvegarde à la fin de son exécution.

Du fait de ce mécanisme, la commande ne peut pas être utilisée depuis les méthodes base Sur démarrage sauvegarde ou Sur arrêt sauvegarde.

---

*4D Server* Lorsqu’elle est appelée depuis un poste client, SAUVEGARDER est considérée comme une procédure stockée, elle est toujours exécutée sur le serveur.

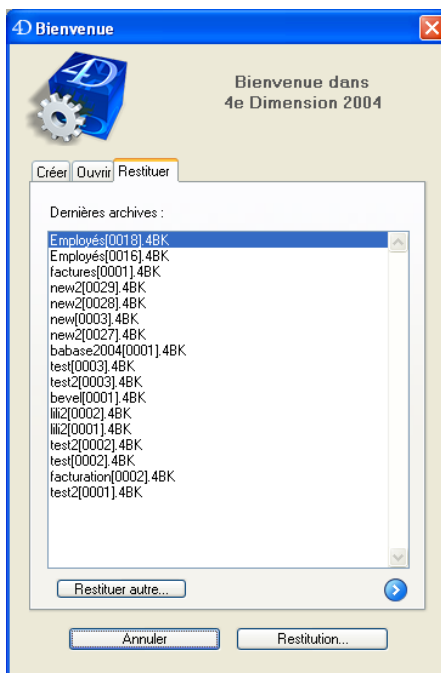
---

Si la sauvegarde se déroule correctement, la variable système OK prend la valeur 1, sinon elle prend la valeur 0. En cas d’incident, une erreur est générée, que vous pouvez intercepter à l’aide d’une méthode de gestion d’erreurs installée à l’aide de la commande APPELER SUR ERREUR.

**RESTITUER****RESTITUER****Paramètres      Type                      Description**

Cette commande ne requiert pas de paramètre

La commande RESTITUER provoque l'affichage de la page de restitution de la boîte de dialogue de bienvenue de 4<sup>e</sup> Dimension :



L'utilisateur peut alors sélectionner une archive à restituer.

Cette commande est utile dans le cadre d'interfaces personnalisées pour la gestion des sauvegardes.

**LIRE INFORMATION SAUVEGARDE****LIRE INFORMATION SAUVEGARDE(sélecteur; info1; info2)****Paramètres      Type                      Description**

sélecteur	Entier long	→	Type d'information à récupérer
info1	Date   Entier	←	Valeur 1 du sélecteur
info2	Heure   Chaîne	←	Valeur 2 du sélecteur

La commande LIRE INFORMATION SAUVEGARDE permet de récupérer des informations relatives à la dernière sauvegarde effectuée sur les données de la base.

Passez dans le paramètre *sélecteur* le type d'information à récupérer. Vous pouvez utiliser une des constantes suivantes, placées dans le thème "Sauvegarde" :

Constante	Type	Valeur
Date dernière sauvegarde	Entier long	0
Statut dernière sauvegarde	Entier long	2
Prochaine sauvegarde	Entier long	4

Le type et le contenu des paramètres *info1* et *info2* dépendent de la valeur de *sélecteur*.

- Si *sélecteur* = 0 (Date dernière sauvegarde), *info1* retourne la date et *info2* l'heure de la dernière sauvegarde.
- Si *sélecteur* = 2 (Statut dernière sauvegarde), *info1* retourne le numéro et *info2* le texte du statut de la dernière sauvegarde.
- Si *sélecteur* = 4 (Prochaine sauvegarde), *info1* retourne la date et *info2* l'heure de la prochaine sauvegarde prévue.

## LIRE INFORMATION RESTITUTION

LIRE INFORMATION RESTITUTION(*sélecteur*; *info1*; *info2*)

Paramètres	Type	Description
<i>sélecteur</i>	Entier long	→ Type d'information à récupérer
<i>info1</i>	Date   Entier	← Valeur 1 du sélecteur
<i>info2</i>	Heure   Chaîne	← Valeur 2 du sélecteur

La commande LIRE INFORMATION RESTITUTION permet de récupérer des informations relatives à la dernière restitution de la base.

Passez dans le paramètre *sélecteur* le type d'information à récupérer. Vous pouvez utiliser une des constantes suivantes, placées dans le thème "Sauvegarde" :

Constante	Type	Valeur
Date dernière restitution	Entier long	0
Statut dernière restitution	Entier long	2

Le type et le contenu des paramètres *info1* et *info2* dépendent de la valeur de *sélecteur*.

- Si *sélecteur* = 0 (Date dernière restitution), *info1* retourne la date et *info2* l'heure de la dernière restitution de la base.
- Si *sélecteur* = 2 (Statut dernière restitution), *info1* retourne le numéro et *info2* le texte du statut de la dernière restitution de la base.

## Méthode base Sur démarrage sauvegarde

Cette nouvelle méthode base est appelée à chaque fois qu'une sauvegarde de la base est sur le point d'avoir lieu (sauvegarde manuelle, sauvegarde automatique périodique ou via la commande [SAUVEGARDER](#)).

Tous les environnements 4D sont concernés : 4<sup>e</sup> Dimension, 4D Server, 4D Client et 4D Runtime.

La méthode base Sur démarrage sauvegarde permet de contrôler le déclenchement de la sauvegarde. Au sein de la méthode, vous devez passer dans le paramètre \$0 une valeur autorisant ou refusant la sauvegarde :

- si \$0 = 0, la sauvegarde peut être lancée.
- si \$0 # 0, la sauvegarde n'est pas autorisée. L'opération est annulée et une erreur est retournée. Vous pouvez récupérer l'erreur à l'aide de la commande [LIRE INFORMATION SAUVEGARDE](#).

Vous pouvez utiliser cette méthode base pour contrôler les conditions d'exécution de la sauvegarde (utilisateur, date de la dernière sauvegarde, etc.).

Vous devez impérativement déclarer le paramètre \$0 (entier long) dans la méthode base :

**C\_ENTIER LONG(\$0)**

## Méthode base Sur arrêt sauvegarde

Cette nouvelle méthode base est appelée à chaque fois qu'une sauvegarde de la base vient de se terminer. Les causes de l'arrêt de la sauvegarde peuvent être la fin de la copie, l'interruption par l'utilisateur ou une erreur.

Tous les environnements 4D sont concernés : 4<sup>e</sup> Dimension, 4D Server, 4D Client et 4D Runtime.

La méthode base Sur arrêt sauvegarde permet de vérifier que la sauvegarde s'est correctement déroulée. Elle reçoit dans le paramètre \$1 une valeur représentant le statut de la sauvegarde à l'issue de son exécution. Si la sauvegarde s'est terminée normalement, \$1 vaut 0.

Sinon, si la sauvegarde a été stoppée par la [Méthode base Sur démarrage sauvegarde](#) (\$0 # 0), \$1 retourne le code effectivement retourné dans le paramètre \$0. Ce principe vous permet de mettre en place un système de gestion d'erreurs personnalisé.

Vous devez impérativement déclarer le paramètre \$1 (entier long) dans la méthode base :

**C\_ENTIER LONG(\$1)**

### Codes d'erreurs de Sauvegarde

Voici la liste des codes d'erreurs 4<sup>e</sup> Dimension relatifs à l'utilisation des commandes de sauvegarde. A noter que ces codes sont positifs afin de les distinguer des erreurs retournées par le Système (telles que par exemple -34, Le disque est plein) :

- 1000 Une erreur est survenue durant la sauvegarde.
- 1301 La sauvegarde a déjà été démarrée.
- 1303 Le volume de destination est introuvable ou le chemin d'accès a été modifié.
- 1403 Cette base travaille sans fichier d'historique.
- 1405 Changement d'historique impossible.
- 1406 Un ancien historique existe déjà.
- 1407 Le nouvel historique ne peut pas être créé.
- 1408 Le nouvel historique ne peut pas être ouvert.

## XML

4<sup>e</sup> Dimension 2004 propose davantage de commandes et d'outils permettant la prise en charge des structures XML (*eXtensible Markup Language*). Trois axes de développement ont été privilégiés :

- Prise en charge du standard SAX,
- Ajout de commandes permettant d'écrire des structures XML et prise en charge de la notation *XPath*,
- Prise en charge du langage XSL et de *Xpath*.

## Changement de nom des commandes 4D 2003

Suite à l'intégration du standard SAX et dans le but de clarifier l'usage de chaque commande, la plupart des commandes XML présentes dans la version 2003 de 4<sup>e</sup> Dimension ont été renommées. Elles sont désormais préfixées DOM. Leur fonctionnement est inchangé :

Ancien nom de commande (4 <sup>e</sup> Dimension 2003.x)	Nouveau nom de commande (4 <sup>e</sup> Dimension 2004.x)
Analyser source XML	DOM Analyser source XML
Analyser variable XML	DOM Analyser variable XML
Compter elements XML	DOM Compter elements XML
Lire premier element XML	DOM Lire premier element XML
Lire element XML suivant	DOM Lire element XML suivant
Lire element XML	DOM Lire element XML
LIRE NOM ELEMENT XML	DOM LIRE NOM ELEMENT XML
LIRE VALEUR ELEMENT XML	DOM LIRE VALEUR ELEMENT XML
Compter attributs XML	DOM Compter attributs XML
LIRE ATTRIBUT XML PAR INDEX	DOM LIRE ATTRIBUT XML PAR INDEX
LIRE ATTRIBUT XML PAR NOM	DOM LIRE ATTRIBUT XML PAR NOM
LIRE ERREUR XML	LIRE ERREUR XML ( <i>inchangé</i> )
Analyser informations XML	DOM Lire informations XML
FERMER XML	DOM FERMER XML

## Prise en charge du standard SAX

4<sup>e</sup> Dimension 2004 vous permet désormais de gérer des documents XML sans limitation de taille, grâce à la prise en charge du standard XML SAX.

## Que sont DOM et SAX ?

DOM (*Document Object Model*) et SAX (*Simple API XML*) sont deux modes d'analyse différents des documents XML.

- Le mode DOM effectue l'analyse d'une source XML et construit sa structure (son "arbre") en mémoire. De ce fait, l'accès à chaque élément de la source est extrêmement rapide. Cependant, la totalité de l'arbre étant contenu dans la mémoire, le traitement de gros documents XML peut dépasser la capacité de la mémoire et provoquer des erreurs.
- Le mode SAX ne construit pas d'arbre en mémoire. Dans ce mode, des "événements" (tels que le début et la fin d'un élément) sont générés lors de l'analyse de la source. Ce mode autorise l'analyse de documents XML de toute taille, quelle que soit la quantité de mémoire disponible.

Pour plus d'informations sur les standards XML, vous pouvez consulter les sites <http://www.saxproject.org/?selected=event> et <http://www.w3schools.com/xml/>

### Création, ouverture et fermeture des documents XML via SAX

Les commandes SAX travaillent avec des références de documents standard de 4<sup>e</sup> Dimension (*DocRef*, référence de type Heure). Il est donc possible d'utiliser ces commandes conjointement avec les commandes 4D permettant de gérer les documents, par exemple ENVOYER PAQUET ou Ajouter a document.

La création et l'ouverture par programmation de documents XML est effectuée via les commandes Créer document et Ouvrir document. Par la suite, l'utilisation d'une commande XML avec ces documents provoquera la mise en oeuvre automatique des mécanismes XML tels que l'encodage. Par exemple, l'en-tête `<?xml version="1.0" encoding="... encodage ..." standalone = "no ">` sera automatiquement écrit dans le document.

La fermeture d'un document XML doit être effectuée à l'aide de la commande FERMER DOCUMENT. Si des éléments XML étaient ouverts, ils sont automatiquement refermés.

### SAX ECRIRE OPTIONS XML

SAX ECRIRE OPTIONS XML(document; encodage{; autonome{; indentation}})

Paramètres	Type	Description
document	docRef	→ Référence du document ouvert
encodage	Alpha	→ Jeu de caractères du document XML
autonome	Booléen	→ Vrai=le document est autonome Faux (défaut)=le document n'est pas autonome
indentation	Booléen	→ Vrai (défaut)=le document est indenté Faux=le document n'est pas indenté

La commande SAX ECRIRE OPTIONS XML initialise le document XML référencé par *document* à l'aide des valeurs passées en paramètres. Ces paramètres permettent de déterminer l'encodage, l'attribut autonome (*standalone*) et l'indentation du document.

- *encodage* : indique le jeu de caractères employé dans le document. Par défaut (si la commande n'est pas appelée), le jeu de caractères UTF-8 (Unicode compressé) est utilisé.



- *autonome* : indique si le document est autonome (Vrai) ou s'il dépend, pour son fonctionnement, d'autres fichiers ou de ressources externes (Faux). Par défaut (si la commande n'est pas appelée ou si le paramètre est omis), le document n'est pas autonome.
- *indentation* : indique si le document doit faire apparaître des indentations (Vrai) ou non (Faux) correspondant aux niveaux hiérarchiques des clés XML. Par défaut (si la commande n'est pas appelée ou si le paramètre est omis), le document est indenté.

Cette commande doit être appelée une seule fois par document et avant la première commande d'écriture XML dans le document, sinon une erreur est générée.

- ▼ Le code suivant :

**SAX ECRIRE OPTIONS XML**(\$RefDoc;"UTF-16";Vrai)

... inscrira cette ligne dans le document :

```
<?xml version="1.0" encoding="UTF-16" standalone="yes"?>
```

## SAX AJOUTER DOCTYPE XML

SAX AJOUTER DOCTYPE XML (document; docType)

Paramètres	Type	Description
document	docRef	→ Référence du document ouvert
docType	Chaîne	→ DocType à ajouter

La commande SAX AJOUTER DOCTYPE XML ajoute une instruction DocType dans le document XML référencé par *document*.

L'instruction DocType permet d'indiquer le type de XML dans lequel le document a été écrit et de désigner la Déclaration de type de document (DTD) utilisée. Une instruction DocType est généralement de la forme `<!DOCTYPE type_XML "adresse_DTD">`

Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0. En cas d'erreur, la commande retourne une erreur qui peut être interceptée via une méthode d'appel sur erreur.

- ▼ L'instruction suivante :

```
vDocType := "Livres SYSTEM \"Livres.DTD\""  
SAX AJOUTER DOCTYPE XML ($RefDoc;vDocType)
```

... inscrira cette ligne dans le document :

```
<!DOCTYPE Livres SYSTEM "Livres.DTD">
```

## SAX AJOUTER COMMENTAIRE XML

SAX AJOUTER COMMENTAIRE XML (document; commentaire)

Paramètres	Type	Description
document	docRef	→ Référence du document ouvert
commentaire	Chaîne	→ Commentaire à ajouter

La commande SAX AJOUTER COMMENTAIRE XML ajoute un commentaire dans le document XML référencé par *document*.

Un commentaire XML est un texte dont le contenu ne sera pas analysé par l'interpréteur XML. Les commentaires XML doivent être encadrés par les caractères <!-- et -->.

Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0. En cas d'erreur, la commande retourne une erreur qui peut être interceptée via une méthode d'appel sur erreur.

▼ L'instruction suivante :

```
vCommentaire := "Créé par 4e Dimension"
```

```
SAX AJOUTER COMMENTAIRE XML ($RefDoc;vCommentaire)
```

... inscrira cette ligne dans le document :

```
<!--Créé par 4e Dimension-->
```

## SAX OUVRIR ELEMENT XML

SAX OUVRIR ELEMENT XML (document; balise {; nomAttribut; valeurAttribut}...{; nomAttributN;valeurAttributN})

Paramètres	Type	Description
document	docRef	→ Référence du document ouvert
balise	Chaîne	→ Nom de l'élément à ouvrir
nomAttribut	Chaîne	→ Nom d'attribut
valeurAttribut	Chaîne	→ Valeur d'attribut

La commande SAX OUVRIR ELEMENT XML permet d'ajouter un nouvel élément dans le document XML référencé par *document* ainsi que, facultativement, des attributs et leurs valeurs.

L'élément ajouté est "ouvert" dans le document (la balise de fin n'est pas ajoutée). Pour refermer un élément créé à l'aide de cette commande, vous devez soit :

- utiliser la commande **SAX FERMER ELEMENT XML**
- refermer le document XML. Dans ce cas, 4<sup>e</sup> Dimension ajoute automatiquement les balises XML de fermeture nécessaires.

Passez dans *balise* le nom de l'élément à créer. Ce nom peut contenir uniquement des lettres, des chiffres, ainsi que les caractères ".", "-", "\_" et ":". Si un caractère invalide est passé dans *balise*, une erreur est générée.

Facultativement, la commande permet de passer un ou plusieurs couple(s) attribut/valeur (sous forme de variables, champs ou valeur littérales) via les paramètres *nomAttribut* et *valeurAttribut*. Vous pouvez passer autant de couples attribut/valeur que vous voulez.

▼ L'instruction suivante :

```
vElement := "Book"
SAX OUVRIR ELEMENT XML($RefDoc;vElement)
```

... inscrira cette ligne dans le document :  
<Book

## SAX OUVRIR ELEMENT XML TABLEAUX

SAX OUVRIR ELEMENT XML TABLEAUX (document; balise{; tabNomsAttributs; tabValeursAttributs}...{;tabNomsAttributsN ;tabValeursAttributsN})

Paramètres	Type	Description
document	docRef	→ Référence du document ouvert
balise	Chaîne	→ Nom de l'élément à ouvrir
tabNomsAttributs	Tab Chaîne	→ Tableau de noms d'attributs
tabValeursAttributs	Tab Chaîne	→ Tableau de valeurs d'attributs

La commande SAX OUVRIR ELEMENT XML TABLEAUX permet d'ajouter un nouvel élément dans le document XML référencé par *document* ainsi que, facultativement, des attributs et leurs valeurs sous forme de tableaux.

Hormis la prise en charge de tableaux (cf. ci-dessous), cette commande est identique à [SAX OUVRIR ELEMENT XML](#). Reportez-vous à la description de cette commande pour le détail de son fonctionnement.

Facultativement, la commande SAX OUVRIR ELEMENT XML TABLEAUX permet de passer plusieurs couples d'attributs et de valeurs d'attributs sous forme de tableaux dans les paramètres *tabNomsAttributs* et *tabValeursAttributs*.

Les tableaux doivent avoir été créés au préalable et fonctionner par paires. Vous pouvez passer autant de couples de tableaux et autant d'éléments dans chaque couple que vous voulez.

- ▼ La méthode suivante :

```

TABLEAU ALPHA(80;tNomsAtt;2)
TABLEAU ALPHA(80;tValeursAtt;2)
vElement:="Book"
tNomsAtt{1}:="Font"
tValeursAtt{1}:="arial"
tNomsAtt{2}:="Style"
tValeursAtt{2}:="Bold"
SAX OUVRIR ELEMENT XML TABLEAUX ($RefDoc;vElement;tNomsAtt;
tValeursAtt)
    
```

... inscrira dans le document :  
 <Book Font="arial" Style="Bold">

## SAX FERMER ELEMENT XML

SAX FERMER ELEMENT XML (document)

Paramètres	Type	Description
document	docRef	→ Référence du document ouvert

La commande **SAX FERMER ELEMENT XML** inscrit dans le document XML référencé par *document* les instructions nécessaires à la fermeture du dernier élément ouvert via la commande **SAX OUVRIR ELEMENT XML**.

L'emploi de cette commande est facultatif. En effet, 4<sup>e</sup> Dimension ajoute automatiquement si nécessaire, au moment de la fermeture des documents XML, les balises de fin d'éléments non refermés explicitement.

- ▼ Si le dernier élément ouvert est <Book>, l'instruction suivante :

```

SAX FERMER ELEMENT XML($RefDoc)
    
```

... inscrira cette ligne dans le document :  
 </Book>

## SAX AJOUTER VALEUR ELEMENT XML

SAX AJOUTER VALEUR ELEMENT XML (document; données)

Paramètres	Type	Description
document	docRef	→ Référence du document ouvert
données	Texte   BLOB	→ Texte ou BLOB à insérer dans le document

La commande SAX AJOUTER VALEUR ELEMENT XML ajoute directement dans le document XML référencé par *document* des *données* de type texte ou BLOB sans les convertir. Cette commande équivaut par exemple à insérer une pièce jointe dans le corps (*body*) d'un email.

Si vous souhaitez encoder le contenu de *données*, vous devez utiliser la commande **ENCODER**. Dans ce cas bien entendu, vous devez passer un BLOB dans *données*.

Pour que cette commande fonctionne, un élément doit être ouvert. Dans le cas contraire, une erreur est générée. Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0.

- ▼ Cet exemple insère le fichier *whitepaper.pdf* dans l'élément XML ouvert :

```
C_BLOB(vBMonBLOB)
DOCUMENT VERS BLOB ("c:\\whitepaper.pdf";vBMonBLOB)
SAX AJOUTER VALEUR ELEMENT XML($RefDoc;vBMonBLOB)
```

## SAX AJOUTER CDATA XML

SAX AJOUTER CDATA XML (document; données)

Paramètres	Type	Description
document	docRef	→ Référence du document ouvert
données	Texte   BLOB	→ Texte ou BLOB à insérer dans le document entre balises CDATA

La commande SAX AJOUTER CDATA XML ajoute dans le document XML référencé par *document* des *données* de type texte ou BLOB. Ces *données* seront automatiquement encadrées par les balises `<CDATA>` et `</CDATA>`. Le texte compris dans une section CDATA est ignoré par l'interpréteur XML.

Si vous souhaitez encoder le contenu de *données*, vous devez utiliser la commande **ENCODER**. Dans ce cas bien entendu, vous devez passer un BLOB dans *données*.

Pour que cette commande fonctionne, un élément doit être ouvert. Dans le cas contraire, une erreur est générée. Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0.

- ▼ Vous souhaitez insérer les lignes suivantes dans votre document XML :

```
function matchwo(a,b)
{
if (a < b && a < 0) then
  {
    return 1
  }
else
  {
    return 0
  }
}
```

Pour cela, il vous suffit d'exécuter le code suivant :

```
C_TEXTE (vtMontexte)
... ` placez ici le texte dans la variable vtMontexte
SAX AJOUTER CDATA XML($RefDoc;vtMontexte)
```

Le résultat sera alors :

```
<![CDATA[
function matchwo(a,b)
{
if (a < b && a < 0) then
  {
    return 1
  }
else
  {
    return 0
  }
}
]]>
```

## SAX AJOUTER INSTRUCTION DE TRAITEMENT

SAX AJOUTER INSTRUCTION DE TRAITEMENT(document; instruction)

Paramètres	Type	Description
document	docRef	→ Référence du document ouvert
instruction	Texte	→ Texte ou BLOB à insérer dans le document

La commande SAX AJOUTER INSTRUCTION DE TRAITEMENT ajoute dans le document XML référencé par *document* une *instruction* de traitement XML.

Une instruction de traitement permet d'indiquer le type d'application et éventuellement des paramètres additionnels permettant de traiter une entité externe non analysable.

La commande formate les données d'instruction conformément au XML. En revanche, les instructions elles-mêmes ne sont pas analysées, il revient au développeur de s'assurer qu'elles sont valides.

Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0 et une erreur est générée.

▼ Le code suivant :

```
vtInstruct:="xml-stylesheet type="+Caractere(Guillemets)+"text/xsl"+
          Caractere (Guillemets)+ "href="+Caractere (Guillemets)+
          "style.xsl"+Caractere (Guillemets)
SAX AJOUTER INSTRUCTION DE TRAITEMENT ($RefDoc;vtInstruct)
```

... inscrira cette ligne dans le document :

```
<?xml-stylesheet type="text/xsl" href="style.xsl"?>
```

## SAX Lire noeud XML

SAX Lire noeud XML (document) → Entier long

Paramètres	Type	Description
document	docRef	→ Référence du document ouvert
Résultat	Entier long	← Événement retourné par la fonction

Cette commande retourne un entier long indiquant le type d'événement SAX retourné durant l'analyse du document XML référencé par *document*.

Les événements pouvant être retournés sont fournis sous forme de constantes dans le thème "XML" :

Constante	Type	Valeur
Début document XML	Entier long	1
Commentaire XML	Entier long	2
Instruction de traitement XML	Entier long	3
Début élément XML	Entier long	4
Fin élément XML	Entier long	5
Donnée XML	Entier long	6
CDATA XML	Entier long	7
Entité XML	Entier long	8
Fin document XML	Entier long	9

Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0 et une erreur est générée.

▼ Exemple de traitement des événements :

```
RefDoc:=Ouvrir document("")
Si (OK=1)
  Repeter
    MonÉvénement:=SAX Lire noeud XML(RefDoc)
  Au cas ou
    : (MonÉvénement=Début document XML)
      FaireQuelqueChose
    : (MonÉvénement=Commentaire XML)
      FaireAutreChose
  Fin de cas
  Jusque (MonÉvénement=Fin document XML)
Fin de si
FERMER DOCUMENT (RefDoc)
```



## SAX LIRE VALEURS DOCUMENT XML

SAX LIRE VALEURS DOCUMENT XML(document; encodage; version, autonome)

Paramètres	Type	Description
document	docRef	→ Référence du document ouvert
encodage	Chaîne	← Jeu de caractères du document XML
version	Chaîne	← Version du XML
autonome	Booléen	← Vrai=le document est autonome, sinon Faux

La commande SAX LIRE VALEURS DOCUMENT XML extrait des informations élémentaires de l'en-tête XML du document XML référencé par *document*.

La commande retourne respectivement le type d'encodage, la version et la propriété "autonome" du document dans les paramètres *encodage*, *version* et *autonome*. Cette commande doit être utilisée dans le contexte de l'événement SAX Début document XML.

Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0 et une erreur est générée.

## SAX LIRE COMMENTAIRE XML

SAX LIRE COMMENTAIRE XML(document; commentaire)

Paramètres	Type	Description
document	docRef	→ Référence du document ouvert
commentaire	Chaîne	← Commentaire XML

Cette commande retourne un *commentaire* si un événement SAX de type Commentaire XML est généré dans le document XML référencé par *document*.

Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0 et une erreur est générée.

## SAX LIRE INSTRUCTION DE TRAITEMENT XML

SAX LIRE INSTRUCTION DE TRAITEMENT XML(document; nom; valeur)

Paramètres	Type	Description
document	docRef	→ Référence du document ouvert
nom	Chaîne	← Nom de l'instruction
valeur	Chaîne	← Valeur de l'instruction

Cette commande retourne le *nom* et la *valeur* de l'instruction de traitement XML analysée dans le document XML référencé par *document*. Cette commande doit être appelée dans le contexte d'un événement Instruction de traitement XML.

- ▼ Considérons l'extrait de code XML suivant :
 

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Edited with XML Spy v3.0.7 NT (http://www.xmlspy.com) by
Myself (4D SA)-->

<?PI TextProcess?>
<!DOCTYPE RootElement SYSTEM "ParseTest.dtd">
```

L'instruction :

**SAX LIRE INSTRUCTION DE TRAITEMENT XML(\$RefDoc;vNom;vValeur)**

... retournera "PI" dans *vNom* et "TextProcess" dans *vValeur*.

## SAX LIRE ELEMENT XML

SAX LIRE ELEMENT XML(document; nom; préfixe; nomsAttributs; valeursAttributs)

Paramètres	Type	Description
document	docRef	→ Référence du document ouvert
nom	Chaîne	← Nom de l'élément
préfixe	Chaîne	← Espace de nommage
nomsAttributs	Tab Alpha	← Noms des attributs
valeursAttributs	Tab Alpha	← Valeurs des attributs

Cette commande retourne diverses informations sur l'élément *nom* présent dans le document XML référencé par *document*. Elle doit être appelée dans le contexte d'un événement SAX Début élément XML ou Fin élément XML. Dans le cas particulier d'un Fin élément XML, les paramètres d'attributs ne sont pas gérés.

*nom* contient le nom de l'élément.

*préfixe* retourne l'espace de nommage (*namespace*) de l'élément. Ce paramètre est vide si aucun espace de nommage n'est associé à l'élément.

La commande remplit le tableau *nomsAttributs* avec les noms des attributs de l'élément cible. Si nécessaire, la commande crée et dimensionne automatiquement le tableau.

La commande remplit également le tableau *valeursAttributs* avec les valeurs des attributs de l'élément cible. Si nécessaire, la commande crée et dimensionne automatiquement le tableau.

Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0 et une erreur est générée.

- ▼ Considérons l'extrait de code XML suivant :

```
<RootElement>
  <Child Att1="111" Att2="222" Att3="333">MonTexte</Child>
</RootElement>
```

Une fois l'instruction suivante exécutée :

**SAX LIRE ELEMENT XML** (RefDoc;vNom;vPréfixe;tAttrNoms;tAttrValeurs)

*...vNom* contiendra "Child"

*vPréfixe* contiendra ""

*tAttrNoms{1}* contiendra "Att1", *tAttrNoms{2}* contiendra "Att2",

*tAttrNoms{3}* contiendra "Att3"

*tAttrValeurs{1}* contiendra "111", *tAttrValeurs{2}* contiendra "222"

*tAttrValeurs{3}* contiendra "333"

## SAX LIRE VALEUR ELEMENT XML

SAX LIRE VALEUR ELEMENT XML(document;valeur)

Paramètres	Type	Description
document	docRef	→ Référence du document ouvert
valeur	Texte   BLOB	← Valeur de l'élément

La commande SAX LIRE VALEUR ELEMENT XML permet de récupérer la *valeur* d'un élément XML existant dans le document XML référencé par *document*. Elle doit être appelée dans le contexte d'un événement SAX Donnée XML.

Passez dans le paramètre *valeur* une variable de type Texte ou BLOB devant récupérer les données. Si vous passez un BLOB, le texte sera retourné tel quel (il ne sera pas modifié).

Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0 et une erreur est générée.

- ▼ Considérons l'extrait de code XML suivant :

```
<RootElement>  
  <Child Att1="111" Att2="222" Att3="333">MonTexte</Child>  
</RootElement>
```

L'instruction :

**SAX LIRE VALEUR ELEMENT XML**(RefDoc;vValeur)

... retournera "MonTexte" dans *vValeur*.

## SAX LIRE CDATA XML

SAX LIRE CDATA XML(document;valeur)

Paramètres	Type	Description
document	docRef	→ Référence du document ouvert
valeur	BLOB	← Valeur de l'élément

La commande SAX LIRE CDATA XML permet de récupérer la *valeur* CDATA d'un élément XML existant dans le document XML référencé par *document*. Elle doit être appelée dans le contexte d'un événement SAX CDATA XML.

Les données sont retournées telles quelles (elles ne sont pas modifiées).

Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0 et une erreur est générée.

- ▼ Considérons l'extrait de code XML suivant :

```
<RootElement>  
  <Child>MonTexte<![CDATA[MonCData]]</Child>  
</RootElement>
```

Le code 4D suivant :

**C\_BLOB** (vDonnées)

**C\_TEXTE** (vDonnéesTexte)

**SAX LIRE CDATA XML**(RefDoc;vDonnées)

vDonnéesTexte:=**BLOB vers texte**(vDonnées;Chaîne en C)

... retournera "MonCData" dans *vDonnéesTexte*.

## SAX LIRE ENTITE XML

SAX LIRE ENTITE XML(document; nom; valeur)

Paramètres	Type	Description
document	docRef	→ Référence du document ouvert
nom	Chaîne	← Nom de l'entité
valeur	Chaîne	← Valeur de l'entité

La commande SAX LIRE ENTITE XML permet de récupérer la *nom* et *valeur* d'une entité XML présente dans le document XML référencé par *document*. Elle doit être appelée dans le contexte d'un événement SAX Entité XML.

Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0 et une erreur est générée.

▼ Considérons l'extrait de code XML suivant :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE body [
  <!ELEMENT body (element*)>
  <!ELEMENT element (#PCDATA)>
  <!ENTITY nom "Le remplacement">
]>
<body>
  <element>L'entité est mise à jour par &nom;</element>
</body>
```

L'instruction :

**SAX LIRE ENTITE XML**(RefDoc;vNom;vValeur)

... retournera "nom" dans *vNom* et "Le remplacement" dans *vValeur*.

## Ecriture de XML via les commandes DOM

4<sup>e</sup> Dimension 2004 propose un ensemble de nouvelles commandes DOM permettant notamment d'écrire et de modifier des arbres XML dans ce standard.

---

*Note* Pour plus d'informations sur le standard DOM et son alternative le standard SAX, reportez-vous au [paragraphe "Que sont DOM et SAX ?"](#), page 375.

---

## Prise en charge de la notation XPath

Deux nouvelles commandes DOM ([DOM Créer element XML](#) et [DOM Chercher element XML](#)) acceptent la notation XPath pour l'accès aux éléments XML.

La *notation* XPath est issue du langage XPath, consacré à la navigation à l'intérieur des structures XML.

La notation XPath permet de désigner directement des éléments au sein d'une structure XML sans devoir indiquer le chemin complet pour y parvenir. Soit par exemple la structure suivante :

```
<RootElement>
  <Elem1>
    <Elem2>
      <Elem3 Font=Verdana Size=10> </Elem3>
    </Elem2>
  </Elem1>
</RootElement>
```

La notation XPath permet d'accéder à l'élément 3 via la syntaxe `/RootElement/Elem1/Elem2/Elem3`.

4<sup>e</sup> Dimension 2004 accepte également les éléments XPath **indexés**, avec la syntaxe `Elément[NumÉlément]`. Soit par exemple la structure suivante :

```
<RootElement>
  <Elem1>
    <Elem2>aaa</Elem2>
    <Elem2>bbb</Elem2>
    <Elem2>ccc</Elem2>
  </Elem1>
</RootElement>
```

La notation XPath permet d'accéder à la valeur "ccc" via la syntaxe `/RootElement/Elem1/Elem2[3]`.

---

*Note* Dans 4<sup>e</sup> Dimension 2004, le paramètre *xChemin* spécifié dans les commandes [DOM Créer élément XML](#) et [DOM Chercher élément XML](#) doit débiter par le caractère / suivi du nom de l'élément courant.

---

Pour une illustration de la notation XPath, reportez-vous aux exemples des commandes [DOM Créer élément XML](#) et [DOM Chercher élément XML](#).

## DOM ECRIRE OPTIONS XML

DOM ECRIRE OPTIONS XML(refElément;encodage{; autonome{; indentation}})

Paramètres	Type	Description
refElément	Alpha	→ Référence d'élément XML
encodage	Alpha	→ Jeu de caractères du document XML
autonome	Booléen	→ Vrai=le document est autonome Faux (défaut)=le document n'est pas autonome
indentation	Booléen	→ Vrai (défaut)=le document est indenté Faux=le document n'est pas indenté

La commande DOM ECRIRE OPTIONS XML permet de définir diverses options qui seront utilisées pour la création de l'arbre XML désigné par *refElément*. Ces options concernent l'encodage, l'attribut autonome (*standalone*) et l'indentation de l'arbre :

- *encodage* : indique le jeu de caractères employé. Par défaut (si la commande n'est pas appelée), le jeu de caractères UTF-8 (Unicode compressé) est utilisé.
- *autonome* : indique si l'arbre est autonome (Vrai) ou s'il dépend, pour son fonctionnement, de ressources externes (Faux). Par défaut (si la commande n'est pas appelée ou si le paramètre est omis), l'arbre n'est pas autonome.
- *indentation* : indique si l'arbre doit faire apparaître des indentations (Vrai) ou non (Faux) correspondant aux niveaux hiérarchiques des clés XML. Par défaut (si la commande n'est pas appelée ou si le paramètre est omis), l'arbre est indenté.
- ▼ Cet exemple définit l'encodage et l'option standalone de l'élément *refElément* :

**DOM ECRIRE OPTIONS XML(refElément;"UTF-16";Vrai)**

**Référence :** [DOM Créer ref XML](#)

**DOM Creer ref XML** DOM Creer ref XML(racine{; nameSpace{; nameSpaceNom; nameSpaceValeur}...{; nameSpaceNomN; nameSpaceValeurN}}) → Chaîne

Paramètres	Type	Description
racine	Alpha	→ Nom de l'élément racine
nameSpace	Chaîne	→ Valeur de l'espace de nommage (Namespace)
nameSpaceNom	Chaîne	→ Nom d'espace de nommage
nameSpaceValeur	Chaîne	→ Valeur d'espace de nommage
Résultat	Chaîne	← Référence de l'élément XML racine

Cette commande crée un arbre XML vide en mémoire et retourne sa référence.

Passez dans le paramètre *racine* le nom de l'élément racine de l'arbre XML.

Passez dans le paramètre facultatif *nameSpace* la déclaration de la valeur de l'espace de nommage (namespace) de l'arbre (par exemple "http://www.4d.com").

Dans ce cas, vous devez préfixer le paramètre *racine* avec le nom de l'espace de nommage, suivi de : (par exemple "MonNameSpace:MaRacine").

---

*Note* L'espace de nommage (*namespace*) est une chaîne de caractères permettant de garantir l'unicité des noms de variables XML. En général, un URL du type http://www.monsite.com/monurl est utilisé. Il n'est pas nécessaire que l'URL soit valide sur le site, il faut juste qu'il soit unique. Pour plus d'informations, reportez-vous au manuel *Mode Structure* de 4<sup>e</sup> Dimension.

---

Vous pouvez déclarer un ou plusieurs espace(s) de nommage supplémentaire(s) dans l'arbre XML généré, à l'aide de couples *nameSpaceNom / nameSpaceValeur*. Vous pouvez passer autant de couples nom / valeur d'espace de nommage que vous voulez.

Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0 et une erreur est générée.

- ▼ Création d'un arbre XML simple :

```
C_ALPHA (16;vRefElem)
vRefElem:=DOM Creer ref XML("MaRacine")
```



Cette méthode produit le résultat suivant :

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<MaRacine/>
```

- ▼ Création d'un arbre XML avec un espace de nommage :

**C\_ALPHA** (16;vRefElem)

\$Racine:="MonNameSpace:MaRacine"

\$Namespace:="http://www.4D.com/tech/namespace"

vRefElem:=**DOM Creer ref XML**(\$Racine;\$Namespace)

Cette méthode produit le résultat suivant :

```
<MonNameSpace:MaRacine xmlns:MonNameSpace="http://
www.4D.com/tech/namespace"/>
```

- ▼ Création d'un arbre XML avec plusieurs espaces de nommage :

**C\_ALPHA** (16;vRefElem)

**C\_ALPHA** (80;\$aNSNom1;\$aNSNom2;\$aNsvaleur1;\$aNsvaleur2)

\$Racine:="MonNameSpace:MaRacine"

\$Namespace:="http://www.4D.com/tech/namespace"

\$aNSNom1:="NSNom1"

\$aNSNom2:="NSNom2"

\$aNsvaleur1:="http://www.4D.com/Prod/namespace"

\$aNsvaleur2:="http://www.4D.com/Mkt/namespace"

vRefElem:=**DOM Creer ref XML**(\$Racine;\$Namespace;\$aNSNom1;  
\$aNsvaleur1;\$aNSNom2;\$aNsvaleur2)

Cette méthode produit le résultat suivant :

```
<MonNameSpace:MaRacine xmlns:MonNameSpace="http://
www.4D.com/tech/nameSpace"
NSNom1="http://www.4D.com/Prod/namespace"
NSNom2="http://www.4D.com/Mkt/namespace"/>
```

## DOM ECRIRE ATTRIBUT XML

DOM ECRIRE ATTRIBUT XML(refElément; nomAttribut; valeurAttribut{;  
nomAttributN; valeurAttributN})

Paramètres	Type	Description
refElément	Alpha	→ Référence d'élément XML
nomAttribut	Chaîne	→ Attribut à définir
valeurAttribut	Chaîne	→ Nouvelle valeur d'attribut

Cette commande permet d'ajouter un ou plusieurs attribut(s) à l'élément XML dont la référence est passée dans le paramètre *refElément*. Elle permet également de définir la valeur de chaque attribut défini.

Passez dans les paramètres *nomAttribut* et *valeurAttribut* respectivement l'attribut à écrire et sa valeur (sous forme de variable, champ ou valeur littérale). Vous pouvez passer autant de couples attribut/valeur que vous voulez.

Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0 et une erreur est générée.

- ▼ Soit la source XML suivante :

```
<Book>
  <Title>The Best Seller</Title>
</Book>
```

Si le code suivant est exécuté :

```
vAttrName:="Font"
vAttrVal:="Verdana"
DOM ECRIRE ATTRIBUT XML(vRefElem;vAttrName;vAttrVal)
```

Nous obtenons :

```
<Book>
  <Title Font=Verdana>The Best Seller</Title>
</Book>
```

## DOM Créer element XML

DOM Créer element XML(refElément; xChemin{; nomAttribut; valeurAttribut}...{; nomAttributN; valeurAttributN}) → Chaîne

Paramètres	Type	Description
refElément	Alpha	→ Référence d'élément XML racine
xChemin	Texte	→ Chemin XPath de l'élément XML à créer
nomAttribut	Chaîne	→ Attribut à définir
valeurAttribut	Chaîne	→ Nouvelle valeur d'attribut
Résultat	Alpha	← Référence de l'élément XML créé

Cette commande permet de créer un nouvel élément dans l'élément XML *refElément* à l'emplacement du noeud désigné par le paramètre *xChemin* et de lui ajouter éventuellement des attributs.

Passez dans *refElément* la référence de l'élément racine (créé par exemple à l'aide de la commande [DOM Créer ref XML](#)).

Passez dans *xChemin* le chemin d'accès au format XML de l'élément à créer. La notation XPath est utilisable dans ce cas (cf. [paragraphe "Prise](#)

en charge de la notation XPath”, page 389). Si des éléments du chemin n’existent pas, ils sont créés.

*Note* Si vous avez défini un ou plusieurs espace(s) de nommage pour l’arbre désigné par *refElement* (cf. [commande DOM Créer ref XML](#), page 392), vous devez préfixer le paramètre *xChemin* du nom de l’espace à utiliser (par exemple “MonNameSpace:MonElement”).

Vous pouvez passer dans les paramètres facultatifs *nomAttribut* et *valeurAttribut* un couple attribut / valeur d’attribut (sous forme de variable, champ ou valeur littérale). Vous pouvez passer autant de couples que vous voulez.

La commande retourne en résultat la référence XML de l’élément créé.

Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0 et une erreur est générée. Une erreur est générée lorsque :

- la référence de l’élément racine n’est pas valide
  - le nom de l’élément à créer n’est pas valide (par exemple, s’il débute par un chiffre).
- ▼ Nous souhaitons créer l’élément suivant :
- ```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<RootElement>
  <Elem1>
    <Elem2>
      <Elem3> </Elem3>
    </Elem2>
  </Elem1>
</RootElement>
```

Pour cela, il suffit d’écrire :

```
C_ALPHA(16;vRefRacine;vRefElement)
vRefRacine:=DOM Créer ref XML("RootElement")
vxPath:="/RootElement/Elem1/Elem2/Elem3"
vRefElement:=DOM Créer element XML(vRefRacine;vxPath)
```

- ▼ Nous souhaitons créer l’élément suivant (comportant des attributs) :
- ```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<RootElement>
  <Elem1>
    <Elem2>
      <Elem3 Font=Verdana Size=10> </Elem3>
    </Elem2>
```

```
</Elem1>  
</RootElement>
```

Pour cela, il suffit d'écrire :

```
C_ALPHA(16;vRefRacine;vRefElement)  
C_ALPHA(80;$aAttrNom1;$aAttrNom2;$aAttrVal1;$aAttrVal2)
```

```
$aAttrNom1:="Font"  
$aAttrNom2:="Size"  
$aAttrVal1:="Verdana"  
$aAttrVal2:="10"
```

```
vRefRacine:=DOM Creer ref XML("RootElement")  
vxPath:="/RootElement/Elem1/Elem2/Elem3"  
vRefElement:=DOM Creer element XML(vRefRacine;vxPath;$aAttrNom1;  
$aAttrVal1;$aAttrNom2;$aAttrVal2)
```

## DOM ECRIRE NOM ELEMENT XML

DOM ECRIRE NOM ELEMENT XML(refElément; nomElément)

Paramètres	Type	Description
refElément	Alpha	→ Référence d'élément XML
nomElément	Alpha	→ Nouveau nom de l'élément

Cette commande permet de modifier le nom de l'élément désigné par *refElément*.

Passez dans *refElément* la référence de l'élément à renommer et dans *nomElément* le nouveau nom de l'élément. Bien entendu, la commande se charge de modifier les balises d'ouverture et de fermeture de l'élément.

Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0 et une erreur est générée. Une erreur est générée lorsque :

- la référence de l'élément n'est pas valide
- le nouveau nom d'élément n'est pas valide (par exemple, s'il débute par un chiffre).

▼ Soit la source XML suivante :

```
<Book>  
  <Title>The Best Seller</Title>  
</Book>
```

Si le code suivant est exécuté, en admettant que *vRefElem* contienne la référence de l'élément 'Book' :

**DOM ECRIRE NOM ELEMENT XML**(*vRefElem*;"BestSeller")

Nous obtenons :

```
<BestSeller>
  <Title>The Best Seller</Title>
</BestSeller>
```

## DOM ECRIRE VALEUR ELEMENT XML

DOM ECRIRE VALEUR ELEMENT XML(*refElément*; *valeurElément*)

Paramètres	Type	Description
<i>refElément</i>	Alpha	→ Référence d'élément XML
<i>valeurElément</i>	Alpha   Variable	→ Nouvelle valeur de l'élément

Cette commande permet de modifier la valeur de l'élément désigné par *refElément*.

Passez dans *valeurElément* une chaîne ou une variable (ou un champ) contenant la nouvelle valeur de l'élément :

- si vous passez une chaîne, la valeur sera affectée telle quelle dans la structure XML.
- si vous passez une variable ou un champ, 4<sup>e</sup> Dimension appliquera un traitement approprié à la valeur en fonction du type de *valeurElément*. Tous les types de données peuvent être utilisés, à l'exception des tableaux, images et pointeurs.

Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0 et une erreur est générée (par exemple, si la référence de l'élément n'est pas valide).

- ▼ Soit la source XML suivante :

```
<Book>
  <Title>The Best Seller</Title>
</Book>
```

Si le code suivant est exécuté, en admettant que *vRefElem* contienne la référence de l'élément 'Title' :

**DOM ECRIRE VALEUR ELEMENT XML**(*vRefElem*;"The Loser")

Nous obtenons :

```
<Book>
  <Title>The Loser</Title>
</Book>
```

## DOM SUPPRIMER ELEMENT XML

DOM SUPPRIMER ELEMENT XML(*refElément*)

Paramètres	Type	Description
<i>refElément</i>	Alpha	→ Référence d'élément XML

Cette commande supprime l'élément désigné par *refElément*.

Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0 et une erreur est générée. Une erreur est générée lorsque :

- la référence de l'élément n'est pas valide,
- l'élément est vide.

## DOM EXPORTER VERS FICHER

DOM EXPORTER VERS FICHER(*refElément*; *cheminFichier*)

Paramètres	Type	Description
<i>refElément</i>	Alpha	→ Référence d'élément XML racine
<i>cheminFichier</i>	Texte	→ Chemin d'accès complet du fichier

Cette commande permet de sauvegarder un arbre XML dans un fichier sur disque.

Passez dans *refElément* la référence de l'élément racine à exporter.

Passez dans *cheminFichier* le chemin d'accès complet du fichier d'export à utiliser ou à créer. Si le fichier n'existe pas, il est créé.

Si vous passez uniquement un nom de fichier (sans chemin d'accès), le fichier sera recherché ou créé à côté du fichier de structure.

Si vous passez une chaîne vide (""), une boîte de dialogue standard d'ouverture et de création de fichier apparaît.

Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0 et une erreur est générée. Une erreur est générée lorsque :

- la référence de l'élément n'est pas valide,
  - le chemin d'accès spécifié n'est pas valide,
  - le volume de stockage retourne une erreur (disque plein, etc.).
- ▼ Cet exemple sauvegarde l'arbre *vRefElem* dans le fichier *MonDoc.xml* :

**DOM EXPORTER VERS FICHER**(*vRefElem*;"C:\\dossier\\MonDoc.xml")

## DOM EXPORTER VERS VARIABLE

DOM EXPORTER VERS VARIABLE(*refElément*; *vVarXml*)

Paramètres	Type	Description
<i>refElément</i>	Alpha	→ Référence d'élément XML racine
<i>vVarXml</i>	Texte   BLOB	→ Variable devant recevoir l'arbre XML

Cette commande permet de sauvegarder un arbre XML dans une variable texte ou BLOB.

Passez dans *refElément* la référence de l'élément racine à exporter.

Passez dans *vVarXml* le nom de la variable devant contenir l'arbre XML. Cette variable peut être de type Texte ou BLOB. Vous pouvez choisir le type en fonction des opérations à effectuer par la suite ou de la taille que l'arbre peut atteindre (rappelons que les variables de type Texte sont limitées à 32000 caractères).

Si l'élément est stocké dans une variable Texte, l'encodage utilisé sera Mac Roman.

Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0 et une erreur est générée (par exemple, si la référence de l'élément n'est pas valide).

- ▼ Cet exemple sauvegarde l'arbre *vRefElem* dans une variable texte :

**C\_TEXTE**(*vtMonTexte*)

**DOM EXPORTER VERS VARIABLE**(*vRefElem*;*vtMonTexte*)

## DOM Chercher élément XML

DOM Chercher element XML(*refElément*; *xChemin*) → Chaîne

Paramètres	Type	Description
<i>refElément</i>	Alpha	→ Référence d'élément XML
<i>xChemin</i>	Texte /BLOB	→ Chemin XPath de l'élément à chercher
Résultat	<i>refElément</i>	← Référence de l'élément trouvé (le cas échéant)

Cette commande vous permet de rechercher un élément XML spécifique dans une structure XML. La recherche débute à l'élément désigné par le paramètre *refElément*.

Le noeud XML à chercher est défini par le paramètre *xChemin*. Le format XPath est utilisable (cf. [paragraphe "Prise en charge de la](#)

notation XPath”, page 389). Il est possible d'utiliser des éléments indexés.

*Note* Conformément à la norme XML, la recherche différencie les majuscules et les minuscules.

La commande retourne en résultat la référence XML de l'élément trouvé.

Si la commande a été exécutée correctement, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0 et une erreur est générée. Une erreur est générée lorsque :

- la référence de l'élément n'est pas valide
  - le chemin XPath passé n'est pas valide.
- ▼ Cet exemple permet de rechercher rapidement un élément XML et d'afficher sa valeur :

```
vTrouvé:=DOM Chercher element XML(vRefElem;"/Book[15]/Title")
DOM LIRE VALEUR ELEMENT XML(vTrouvé;valeur)
ALERTE("La valeur de l'élément est : \""+valeur+"\"")
```

- ▼ La même recherche peut également être effectuée ainsi :

```
vTrouvé:=DOM Chercher element XML(vRefElem;"/Book[15]")
vTrouvé:=DOM Chercher element XML(vTrouvé;"/Title")
DOM LIRE VALEUR ELEMENT XML(vTrouvé;valeur)
ALERTE("La valeur de l'élément est : \""+valeur+"\"")
```

## Commande modifiée

### DOM LIRE VALEUR ELEMENT XML

DOM LIRE VALEUR ELEMENT XML(refElément; valeurElément{; cDATA})

Paramètres	Type	Description
refElément	Alpha	→ Référence d'élément XML
valeurElément	Variable	← Valeur de l'élément
cDATA	Variable	← Contenu de la section CDATA

La commande DOM LIRE VALEUR ELEMENT XML accepte un paramètre optionnel supplémentaire, *cDATA*, permettant de récupérer le contenu de la ou des section(s) CDATA de l'élément XML *refElément*.



Comme pour le paramètre *valeurElément*, 4<sup>e</sup> Dimension tentera de convertir la valeur obtenue dans le type de la variable passée en paramètre.

## Codes d'erreurs XML

Voici la liste des codes d'erreurs 4<sup>e</sup> Dimension relatifs à l'utilisation des commandes XML :

### Erreurs SAX

- 9915 La référence du document est incorrecte.
- 9916 L'élément n'est pas ouvert.
- 9917 Le type du tableau passé en paramètre est incorrect.
- 9918 Le nom de l'élément est incorrect.
- 9919 Cet encodage n'est pas supporté.
- 9920 Le type du noeud est incorrect.

### Erreurs DOM

- 9925 L'élément référencé est nul.
- 9926 L'élément référencé est invalide.
- 9927 L'élément référencé n'est pas le "root".
- 9928 Le nom de l'élément est inconnu.
- 9929 L'index pour cet élément est invalide.
- 9930 Il n'existe pas d'attribut de ce nom pour cet élément.
- 9931 L'index pour cet attribut est invalide.
- 9932 La DLL XML n'est pas chargée.
- 9933 Le fichier XML n'est pas valide.
- 9934 Le fichier XML n'est pas bien formé.
- 9935 Le fichier XML n'est pas valide ou n'est pas bien formé.

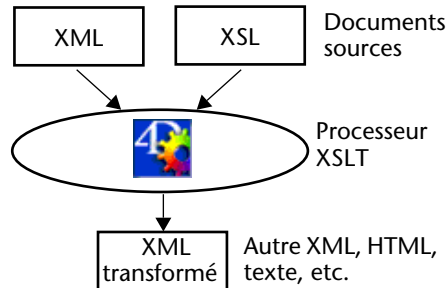
## Prise en charge des transformations XSL

4<sup>e</sup> Dimension 2004 prend en charge l'application de feuilles de style XSL (*eXtended Stylesheet Language*). Le langage XSL permet de modifier le balisage d'un document XML.

Le langage XSL comporte deux facettes :

- **le formatage** : il permet d'appliquer des règles de style et d'affichage pour les éléments XML, un peu à l'image des CSS (*Cascading StyleSheet*) du langage HTML.
- **la transformation** : il permet de transformer un balisage XML en un autre système de balisage, par exemple en HTML. Cette fonction de transformation est plus spécifiquement appelée **XSLT**. Une feuille de style XSL peut entièrement réorganiser les éléments XML d'un

document en les sélectionnant puis en les transformant en d'autres éléments. Cette fonction est utile par exemple pour harmoniser un ensemble de documents XML disparates.



---

*Note* 4<sup>e</sup> Dimension utilise la bibliothèque Xalan-C\_1\_6\_0.dll pour la réalisation des transformations XSL. Xalan est un processeur XSLT du domaine public. Pour plus d'informations, reportez-vous à l'adresse <http://xml.apache.org/xalan-c/index.html>

---

Les feuilles de style XSL sont des documents texte (extension “.xsl”) générés manuellement ou à l'aide d'applications spécialisées. Le langage XSL comporte divers éléments et fonctions permettant d'effectuer tout type de transformation dynamique. Pour plus d'informations sur ce langage, reportez-vous au site <http://xmlfr.org> (par exemple).

4<sup>e</sup> Dimension 2004 vous permet de transformer un document XML à l'aide d'une feuille de style XSL existante (commande **APPLIQUER TRANSFORMATION XSLT**). En outre, 4<sup>e</sup> Dimension vous permet de modifier à la volée des paramètres de la feuille de style XSL via la commande **FIXER PARAMETRE XSLT**.

---

*Note* Une nouvelle option dans la boîte de dialogue d'exportation vous permet d'utiliser une feuille de style XSL lors d'un export XML et donc de générer un document XML transformé (cf. [paragraphe “Appliquer la transformation XSL”](#), page 38).

---

## APPLIQUER TRANSFORMATION XSLT

APPLIQUER TRANSFORMATION XSLT(sourceXML; feuilleXSL; résultat)

Paramètres	Type	Description
sourceXML	Chaîne   BLOB →	Nom ou chemin d'accès du document XML source, ou BLOB contenant le XML source
feuilleXSL	Chaîne   BLOB →	Nom ou chemin d'accès du document contenant la feuille de style XSL, ou BLOB contenant la feuille de style XSL
résultat	Chaîne   BLOB →	Nom ou chemin d'accès du document recevant le résultat de la transformation XSLT, ou BLOB recevant le résultat de la transformation XSLT

La commande APPLIQUER TRANSFORMATION XSLT applique une transformation XSLT à un document ou un BLOB contenant du XML et génère un document ou un BLOB *résultat*. La portée de cette commande est le process courant.

La commande requiert trois paramètres de type BLOB ou chaîne de caractères. Attention, la commande accepte uniquement des variables ou des champs en paramètres.

Si vous passez une chaîne de caractères, vous désignez un document. Dans ce cas, vous pouvez passer uniquement le nom (le document doit se trouver à côté de la structure de la base de données) ou le chemin d'accès complet du document.

Il n'est pas possible de mixer différents types de paramètres au sein d'un même appel.

- Le paramètre *sourceXML* doit contenir le source XML à transformer. La commande vérifie la validité du code XML.
- Le paramètre *feuilleXSL* doit contenir la feuille de style XSL à utiliser pour la transformation XSLT. Cette feuille de style peut avoir été générée manuellement ou via un logiciel spécialisé. La commande vérifie la validité du code XML.
- Le paramètre *résultat* doit contenir le nom du document ou du BLOB devant recueillir le résultat de la transformation XSLT. Si vous passez un nom de document n'existant pas à l'emplacement désigné, 4<sup>e</sup> Dimension le crée automatiquement. Si le document est déjà ouvert en écriture, une erreur est générée.

La commande analyse le source XML et le transforme à l'aide des instructions de la feuille de style XSL. Si la commande **FIXER PARAMETRE XSLT** a été utilisée au préalable, la commande remplace les paramètres définis par leur valeur. Le résultat de la transformation est écrit dans le document ou BLOB *résultat*.

Si la transformation a été correctement effectuée, la variable système **OK** prend la valeur 1, sinon elle prend la valeur 0.

---

*Note* Afin d'optimiser les performances de cette commande notamment en cas d'applications successives de la même feuille XSL, 4<sup>e</sup> Dimension compile et conserve en mémoire les dernières feuilles XSL utilisées.

---

- ▼ Reportez-vous à l'exemple de la commande **FIXER PARAMETRE XSLT**.

**Référence :** **FIXER PARAMETRE XSLT**

## **FIXER PARAMETRE XSLT**

**FIXER PARAMETRE XSLT**(nomParam; valeurParam)

<b>Paramètres</b>	<b>Type</b>	<b>Description</b>
nomParam	Chaîne	→ Nom du paramètre à chercher dans la feuille XSL
valeurParam	Chaîne	→ Valeur du paramètre à utiliser dans le document transformé

La commande **FIXER PARAMETRE XSLT** doit être utilisée conjointement avec la commande **APPLIQUER TRANSFORMATION XSLT**. Elle permet de définir les valeurs de paramètres variables placés dans une feuille de style XSL au moment de la transformation XSLT d'un document XML. A l'aide de cette commande, il est notamment possible d'insérer des valeurs issues de traitements 4<sup>e</sup> Dimension dans la feuille de style XSL, juste avant son utilisation par **APPLIQUER TRANSFORMATION XSLT**.

---

*Note* La portée de cette commande est le process courant. Elle doit être appelée dans le même process que la commande **APPLIQUER TRANSFORMATION XSLT** associée.

---

Passez dans *nomParam* le nom du paramètre XSL variable à remplacer. Ce paramètre doit être présent dans la feuille de style XSL sous la forme \$aremplacer. En revanche, le caractère \$ n'est pas nécessaire dans *nomParam*. Par exemple, si l'instruction `<xsl:template match=$mavar>` est placée dans le fichier XSL, il vous suffira de passer "mavar" dans *nomParam* pour désigner ce paramètre.

Passez dans *valeurParam* la valeur que vous souhaitez insérer dans le fichier transformé en lieu et place du paramètre XSL variable. Pour reprendre l'exemple précédent, si vous passez la valeur "titre" dans *valeurParam*, la transformation XSLT prendra en compte l'instruction `<xsl:template match="titre">` (ce qui désigne les éléments "titre" comme sujets de l'application d'une règle de style).

*Note* Pour une description détaillée du langage XSL, vous pouvez vous reporter aux nombreux sites Web qui lui sont consacrés, par exemple <http://xmlfr.org>.

Pour passer plusieurs paramètres à une feuille de style XSL, il suffit d'appeler plusieurs fois la commande **FIXER PARAMETRE XSLT**. Les paramètres sont "empilés" jusqu'à l'appel de **APPLIQUER TRANSFORMATION XSLT** dans le même process. A l'issue de l'exécution de **APPLIQUER TRANSFORMATION XSLT**, la "pile" de paramètres est automatiquement effacée.

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

- ▼ L'exemple suivant définit deux paramètres XSL puis transforme le document *mondoc.xml* en fichier html via la feuille de style *mafeuille.xml* :

```

FIXER PARAMETRE XSLT("varstyle";"gras")
FIXER PARAMETRE XSLT("varcouleur";"bleu")
$docxml:="mondoc.xml"
$feuillexml:="mafeuille.xml"
$dohtml:="mondoc.html"
APPLIQUER TRANSFORMATION XSLT($docxml;$feuillexml;$dohtml)

```

*Référence* : **APPLIQUER TRANSFORMATION XSLT**

## LIRE ERREUR XSLT

LIRE ERREUR XSLT (texteErreur{; ligne{; colonne{}})

Paramètres	Type	Description
texteErreur	Variable	← Texte de l'erreur
ligne	Variable	← Numéro de ligne
colonne	Variable	← Numéro de colonne

Cette commande retourne dans le paramètre *texteErreur* la description de la dernière erreur éventuellement rencontrée lors de la transformation XSLT effectuée dans le process courant. Les informations retournées sont fournies par la librairie Xerces.dll.

Les paramètres optionnels *ligne* et *colonne* désignent l'emplacement de l'erreur dans le fichier XSL : ils récupèrent respectivement le numéro de la ligne et, dans cette ligne, la position du premier caractère de l'expression à l'origine de l'erreur.

Si la commande a été correctement exécutée, la variable système *OK* prend la valeur 1. Si une erreur se produit, elle prend la valeur 0.

## Web Services (Client)

**FIXER OPTION WEB SERVICE**    FIXER OPTION WEB SERVICE (option; valeur)

Paramètres	Type	Description
option	Entier long	→ Code de l'option à fixer
valeur	Entier long	→ Valeur de l'option

---

*Note* Cette commande est destinée aux utilisateurs avancés des Web Services. Son emploi est facultatif.

---

La commande **FIXER OPTION WEB SERVICE** permet de définir différentes options qui seront utilisées lors de la prochaine requête SOAP déclenchée par la commande **APPELER WEB SERVICE**.

Vous pouvez appeler cette commande autant de fois qu'il y a d'options à fixer avant un **APPELER WEB SERVICE**.

Passez dans le paramètre *option* le numéro de l'option à définir et dans le paramètre *valeur* la nouvelle valeur de l'option.

Pour le paramètre *option*, vous pouvez utiliser une des constantes prédéfinies suivantes, situées dans le thème "Web Services (Client)" :

- Web Service header SOAP
- Web Service timeout HTTP
- Web Service version SOAP

Voici la description des options et des valeurs possibles :

- Web Service header SOAP  
*valeur* = référence d'élément xml racine à insérer en tant que header (en-tête) de la requête SOAP.  
Cette option permet d'insérer un header dans la requête SOAP générée par la commande **APPELER WEB SERVICE**. Par défaut, les requêtes SOAP

ne comportent pas d'en-tête spécifique. Cependant, certains Web Services requièrent la présence de cet en-tête, par exemple pour la gestion de paramètres d'identification.

- **Web Service timeout HTTP**  
*valeur* = "timeout" de la partie cliente exprimé en secondes.  
 Le timeout de la partie cliente est le délai d'attente du client Web Service en cas de non-réponse du serveur. A l'issue de ce délai, le client referme la session, la requête est perdue.  
 Par défaut, ce délai est de 180 secondes. Il peut être modifié en raison de caractéristiques particulières (état du réseau, spécificités du Web Service, etc.).
- **Web Service version SOAP**  
*valeur* = Web Service SOAP\_1\_1 ou Web Service SOAP\_1\_2 (constantes du thème "Web Services (Client)" )  
 Cette option permet de préciser la version du protocole SOAP utilisée dans la requête. Passez dans *valeur* la constante Web Service SOAP\_1\_1 pour indiquer la version 1.1 et la constante Web Service SOAP\_1\_2 pour indiquer la version 1.2.

L'ordre d'appel des options n'a pas d'importance. Si une même *option* est définie plusieurs fois, seule la valeur du dernier appel est prise en compte.

- ▼ Insertion d'un en-tête personnalisé dans la requête SOAP :

```

`Création d'une référence XML
C_ALPHA(16;vRefRacine;vRefElement)
vRefRacine:=DOM Creer ref XML("RootElement")
vxPath:="/RootElement/Elem1/Elem2/Elem3"
vRefElement:=DOM Creer element XML(vRefRacine;vxPath)
`Modification de l'en-tête SOAP avec la référence :
FIXER OPTION WEB SERVICE(Web Service header SOAP;vRefElement)

```

- ▼ Utilisation de la version 1.2 du protocole SOAP :

```

FIXER OPTION WEB SERVICE(Web Service version SOAP ;
Web Service SOAP_1_2)

```

## Source de données externes

Ce nouveau thème contient les commandes permettant à 4<sup>e</sup> Dimension d'accéder à des données stockées dans d'autres applications via des protocoles standard. Dans 4<sup>e</sup> Dimension 2004, seules des commandes ODBC sont disponibles.

Le standard ODBC (Open DataBase Connectivity) définit une librairie de fonctions standard. Ces fonctions permettent à une application telle que 4<sup>e</sup> Dimension d'accéder via le langage SQL à tout système de gestion de données compatible ODBC (bases de données, tableurs, etc.).

---

*Note* 4<sup>e</sup> Dimension 2004 permet également d'importer et d'exporter des données dans une source ODBC en mode Utilisation (cf. [paragraphe "Importation et exportation via une source de données ODBC"](#), page 38).

---

Les commandes ODBC de haut niveau intégrées dans le thème "Source de données externes" de 4<sup>e</sup> Dimension permettent de mettre en place des solutions simples pour faire communiquer les applications 4<sup>e</sup> Dimension et des sources de données ODBC. Si vos applications nécessitent une prise en charge plus étendue du standard ODBC, vous devrez acquérir le nouveau plug-in ODBC "bas niveau" de 4D (cf. note ci-dessous).

**Note aux utilisateurs de 4D ODBC :** Le plug-in existant 4D ODBC n'est pas compatible avec la version 2004 de 4<sup>e</sup> Dimension. 4D fournit un plug-in ODBC de nouvelle génération (4D ODBC Pro) permettant de gérer des connexions ODBC complexes. 4D fournit également un composant spécifique permettant de convertir dans une base les commandes de l'ancien plug-in 4D ODBC en commandes 4D ODBC Pro.

Les commandes ODBC intégrées de 4<sup>e</sup> Dimension 2004 appliquent les principes suivants :

- La portée d'une connexion est le process. Si vous souhaitez gérer plusieurs connexions simultanément, vous devez démarrer un process par **ODBC LOGIN**.

La commande **ODBC ANNULER CHARGEMENT** permet d'exécuter plusieurs requêtes SELECT dans la même connexion.



- Vous pouvez intercepter les erreurs ODBC éventuellement générées lors de l'exécution d'une des commandes ODBC à l'aide la commande APPELER SUR ERREUR. La commande ODBC LIRE DERNIERE ERREUR permet dans ce cas d'obtenir des informations supplémentaires.

## ODBC LOGIN

ODBC LOGIN({nomSource}; utilisateur; motDePasse}})

Paramètres	Type	Description
nomSource	Chaîne	→ Nom de la source de données dans le gestionnaire ODBC
utilisateur	Chaîne	→ Nom d'utilisateur enregistré dans la source de données
motDePasse	Chaîne	→ Mot de passe de l'utilisateur

La commande ODBC LOGIN vous permet d'ouvrir une connexion avec une source de données ODBC externe.

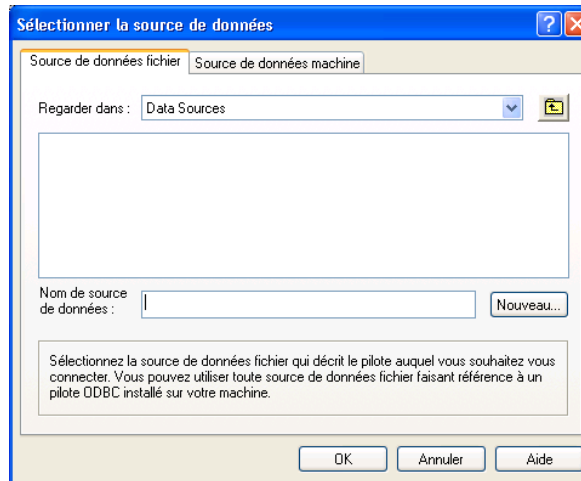
Le paramètre *nomSource* contient le nom de la source de données telle qu'elle a été définie dans le gestionnaire du pilote ODBC.

Le paramètre *utilisateur* contient le nom de l'utilisateur autorisé à se connecter à la source de données externe. Par exemple, avec Oracle®, ce nom d'utilisateur peut être "Scott".

Le paramètre *motDePasse* contient le mot de passe de l'utilisateur autorisé à se connecter. Par exemple, avec Oracle®, ce mot de passe peut être "tiger".

Tous ces paramètres sont facultatifs. Si aucun paramètre n'est passé, la commande provoquera l'affichage de la boîte de dialogue du

gestionnaire ODBC, permettant de désigner manuellement la source de données à laquelle se connecter :



La portée de cette commande est le process. Autrement dit, si vous souhaitez ouvrir deux connexions distinctes, vous devez créer deux process et ouvrir chaque connexion dans chaque process.

Si la connexion est correctement établie, la variable système `OK` prend la valeur 1, sinon elle prend la valeur 0.

- ▼ Cette instruction provoque l’affichage de la boîte de dialogue du gestionnaire ODBC :

### ODBC LOGIN

- ▼ Cette instruction provoque la connexion à la source de données ODBC “MonOracle” avec les identifiants “Scott” et “tiger” :

**ODBC LOGIN**("MonOracle";"Scott";"tiger")

## ODBC LOGOUT

ODBC LOGOUT

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

Cette commande ne requiert pas de paramètre

La commande ODBC LOGOUT referme la connexion avec une source ODBC ouverte dans le process courant (le cas échéant). S’il n’y a pas de connexion, la commande ne fait rien.

Si la connexion a été correctement refermée, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0. Vous pouvez intercepter les éventuelles erreurs à l'aide d'une méthode installée par la commande *APPELER SUR ERREUR*.

## ODBC FIXER OPTION

ODBC FIXER OPTION(option; valeur)

Paramètres	Type	Description
option	Entier long	→ Numéro d'option à définir
valeur	Entier long	→ Nouvelle valeur de l'option

La commande ODBC FIXER OPTION permet de modifier la *valeur* de l'option passée dans le paramètre *option*.

Vous pouvez passer dans *option* l'une des constantes suivantes, placées dans le thème "Source de données externes" :

Constante	Description et valeurs possibles
ODBC Asynchrone (1)	0 = connexion synchrone (valeur par défaut) 1 (ou valeur différente de 0) = connexion asynchrone
ODBC Nombre maxi lignes (2)	Nombre maximum de lignes dans l'ensemble résultant (utilisé pour les prévisualisations)
ODBC Longueur maxi données (3)	Longueur maximale des données retournées
ODBC Timeout requête (4)	Durée maximale d'attente de la réponse lors de l'exécution de la commande <b>ODBC EXECUTER</b> . Cette valeur doit être fixée avant l'ouverture de la connexion pour être prise en compte Valeurs : durée en secondes Par défaut : pas de timeout
ODBC Timeout connexion (5)	Durée maximale d'attente lors de l'exécution de la commande <b>ODBC LOGIN</b> Valeurs possibles : durée en secondes Par défaut : pas de timeout

Si la commande a été correctement exécutée, la variable système *OK* retourne 1, sinon elle retourne 0.

**Référence :** **ODBC LIRE OPTION**

**ODBC LIRE OPTION** ODBC LIRE OPTION(option; valeur)

Paramètres	Type	Description
option	Entier long	→ Numéro d'option
valeur	Entier long	← Valeur de l'option

La commande ODBC LIRE OPTION retourne la *valeur* courante de l'option passée dans le paramètre *option*.

Pour plus d'informations sur les différentes options et leurs valeurs associées, reportez-vous à la description de la [commande ODBC FIXER OPTION](#), page 411.

Si la commande a été correctement exécutée, la variable système OK retourne 1, sinon elle retourne 0.

**Référence :** [ODBC FIXER OPTION](#)

**ODBC EXECUTER** ODBC EXECUTER(instructionSQL{;objetLié1;...;objetLié\_n})

Parameters	Type	Description
instructionSQL	Texte	→ Commande SQL à exécuter
objetLié1...n	Var   Champ	← Réception du résultat (si nécessaire)

La commande ODBC EXECUTER permet d'exécuter une commande SQL et d'associer le résultat à des objets 4D (tableaux, variables ou champs) liés.

Pour que la commande puisse être exécutée, une connexion valide doit être ouverte dans le process courant.

Le paramètre *instructionSQL* contient la commande SQL à exécuter. Le paramètre *objetLié* reçoit les résultats.

Les variables sont liées dans l'ordre de la colonne, ce qui signifie que les éventuelles colonnes distantes supplémentaires sont ignorées.

Si des champs 4<sup>e</sup> Dimension sont passés dans le(s) paramètre(s) *objetLié*, la commande créera des enregistrements et les sauvegardera automatiquement. Les champs doivent appartenir à la même table (il n'est pas possible de passer un champ de la table 1 et un champ de la table 2 dans le même appel). Si des champs de tables différentes sont passés, une erreur est générée.

Si vous passez des tableaux ou des variables 4D dans le(s) paramètre(s) *objetLié*, il est conseillé de les déclarer préalablement à l'appel de la commande afin de contrôler le type de données traitées. Les tableaux sont redimensionnés automatiquement si nécessaire.

Dans le cas d'une variable 4D, un seul enregistrement est récupéré à la fois.

- ▼ Dans cet exemple, nous récupérons la colonne *ename* de la table *emp* dans la source de données externe. Le résultat est stocké dans le champ 4D [Employés]Nom. Les enregistrements 4<sup>e</sup> Dimension seront créés automatiquement :

```
SQLStmt:="SELECT ename FROM emp"
ODBC EXECUTER(SQLStmt;[Employés]Nom)
ODBC CHARGER ENREGISTREMENT(ODBC Tous les enregistrements)
```

- ▼ Pour contrôler la création des enregistrements, il est possible d'inclure le code au sein d'une transaction et de ne la valider que si le déroulement de l'opération s'est avéré satisfaisant :

```
ODBC LOGIN("mysql";"root";"")
SQLStmt:="SELECT alpha_field FROM app_testTable"
DEBUT TRANSACTION
ODBC EXECUTER(SQLStmt;[Table 2]Champ1)
Tant que(Non(ODBC Fin de selection))
  ODBC CHARGER ENREGISTREMENT
  ... `Placer ici le code de validation des données
Fin tant que
VALIDER TRANSACTION `Validation de la transaction
```

- ▼ Dans cet exemple, nous récupérons la colonne *ename* de la table *emp* dans la source de données externe. Le résultat est stocké dans le tableau *tNoms*. Nous récupérons les enregistrements 10 par 10.

```
TABLEAU ALPHA(30;tNoms;20)
SQLStmt:="SELECT ename FROM emp"
ODBC EXECUTER(SQLStmt;tNoms)
Tant que(Non(ODBC Fin de selection))
  ODBC CHARGER ENREGISTREMENT(10)
Fin tant que
```

- ▼ Dans cet exemple, nous récupérons les colonnes *ename* et *job* de la table *emp* pour un ID spécifique (clause WHERE) de la source de données externe. Le résultat est stocké dans les variables 4D *vNom* and *vJob*. Seul le premier enregistrement est récupéré.

```
SQLStmt:="SELECT ename, job FROM emp WHERE id = 3"
ODBC EXECUTER(SQLStmt;vName;vJob)
ODBC CHARGER ENREGISTREMENT
```

- ▼ Dans cet exemple, nous récupérons la colonne *Champ\_Blob* de la table *Test* dans la source de données externe. Le résultat est stocké dans une variable BLOB, dont la valeur est mise à jour à chaque chargement d'enregistrement.

```
C_BLOB(MonBlob)
ODBC LOGIN
ODBC EXECUTER("SELECT Champ_Blob FROM Test";MonBlob)
Tant que(Non(ODBC Fin de selection))
    `On parcourt le résultat
    ODBC CHARGER ENREGISTREMENT
    `La valeur de MonBlob est mise à jour à chaque appel
Fin tant que
```

## ODBC Fin de selection

ODBC Fin de selection → Booléen

Paramètres	Type	Description
		Cette commande ne requiert pas de paramètre

Résultat	Booléen	← Limites de l'ensemble de recherche atteintes
----------	---------	--

La commande ODBC Fin de selection indique si les limites de l'ensemble résultat ont été atteintes.

- ▼ Le code ci-dessous se connecte à une source de données externe (Oracle) à l'aide des paramètres suivants :

```
C_TEXTE(vName)
ODBC LOGIN("TestOracle";"scott";"tiger")
```

```

Si (OK=1)
  ODBC EXECUTER("SELECT ename FROM emp";vName)
  Tant que(Non(ODBC Fin de selection))
    ODBC CHARGER ENREGISTREMENT
  Fin tant que
  ODBC LOGOUT
Fin de si

```

Cet exemple retournera dans la variable 4D *vName* les noms (*ename*) stockés dans la table nommée *emp*.

## ODBC CHARGER ENREGISTREMENT

ODBC CHARGER ENREGISTREMENT{(nombreEnr)}

Paramètres	Type	Description
nombreEnr	Entier	→ Nombre d'enregistrements à charger

La commande ODBC CHARGER ENREGISTREMENT récupère dans 4<sup>e</sup> Dimension un ou plusieurs enregistrement(s) provenant de la source ODBC ouverte dans la connexion courante.

Le paramètre facultatif *nombreEnr* permet de définir le nombre d'enregistrements à récupérer :

- Si vous omettez ce paramètre, la commande récupèrera l'enregistrement courant dans la source de données. Ce principe correspond à la récupération des données dans une boucle où un enregistrement est reçu à la fois.
- Si vous passez une valeur entière dans *nombreEnr*, la commande récupèrera *nombreEnr* enregistrements.
- Si vous passez la constante ODBC Tous les enregistrements (ou la valeur -1), la commande récupèrera tous les enregistrements de la table.

---

*Note* Ces deux derniers paramétrages n'ont de sens que si les données récupérées sont associées à des tableaux ou des champs 4D.

---

## ODBC ANNULER CHARGEMENT

ODBC ANNULER CHARGEMENT

Paramètres	Type	Description
Cette commande ne requiert pas de paramètre		

La commande ODBC ANNULER CHARGEMENT met fin à la requête SELECT courante et initialise les paramètres.

Cette commande permet d'exécuter plusieurs requêtes SELECT au sein d'une même connexion (c'est-à-dire un même curseur) initiée par la commande **ODBC LOGIN**.

- ▼ Dans cet exemple, deux requêtes sont exécutées dans la même connexion :

```
C_BLOB(Monblob)
C_TEXTE(MonTexte)
ODBC LOGIN("mysql","root","")
```

```
SQLStmt:="SELECT blob_field FROM app_testTable"
ODBC EXECUTER(SQLStmt;Monblob)
Tant que(Non(ODBC Fin de selection))
  ODBC CHARGER ENREGISTREMENT
Fin tant que
```

```
`Réinitialisation du curseur
ODBC ANNULER CHARGEMENT
```

```
SQLStmt:="SELECT Name FROM Employee"
ODBC EXECUTER(SQLStmt;MonTexte)
Tant que(Non(ODBC Fin de selection))
  ODBC CHARGER ENREGISTREMENT
Fin tant que
```

## ODBC FIXER PARAMETRE

ODBC FIXER PARAMETRE(objet; typeParam)

Paramètres	Type	Description
objet	Objet 4D	→ Objet 4D à utiliser (variable, tableau ou champ)
typeParam	Entier long	→ Type du paramètre

La commande ODBC FIXER PARAMETRE permet d'utiliser la valeur d'une variable, d'un tableau ou d'un champ 4D dans les requêtes ODBC.

### Utiliser des objets 4<sup>e</sup> Dimension dans les requêtes ODBC

4<sup>e</sup> Dimension propose deux modes d'insertion des objets 4D dans les requêtes ODBC : l'association directe et la définition de paramètres via ODBC FIXER PARAMETRE.

- **Association directe** : dans ce mode, il suffit simplement d'insérer le nom de l'objet 4D à utiliser (variable, tableau ou champ) entre les caractères << et >> dans le texte de la requête. Par exemple : INSERT INTO emp (empno,ename) VALUES (<<vEmpno>>,<<vEname>>)  
Dans cet exemple, les valeurs des variables 4D *vEmpno* et *vEname* seront



substituées aux paramètres lors de l'exécution de la requête. Cette solution fonctionne également avec les champs et les tableaux 4D. Cette syntaxe, simple d'utilisation, présente toutefois l'inconvénient de n'être pas conforme à la norme SQL et de ne pas permettre l'utilisation de paramètres de sortie. Pour y remédier, vous pouvez utiliser la commande ODBC FIXER PARAMETRE (second mode).

- **Définition de paramètres** : dans ce mode, vous utilisez la commande ODBC FIXER PARAMETRE pour définir les objets 4D que vous souhaitez utiliser dans la requête ODBC.
  - passez dans le paramètre *objet* l'objet 4D (variable, tableau ou champ) à utiliser dans la requête.
  - passez dans le paramètre *typeParam* le type SQL du paramètre. Vous pouvez passer une valeur ou utiliser l'une des constantes suivantes, placées dans le thème "Source de données externes" :

Constante	Type	Valeur
OBDC Paramètre entrée	Entier long	1
OBDC Paramètre entrée sortie	Entier long	2
OBDC Paramètre sortie	Entier long	4

La valeur de l'objet 4D est substituée au caractère ? dans la requête SQL (syntaxe standard).

Si la requête comporte plusieurs caractères ?, plusieurs appels à ODBC FIXER PARAMETRE seront nécessaires. Les valeurs des objets 4D seront affectées séquentiellement dans la requête, dans l'ordre d'exécution des commandes.

- ▼ Cet exemple permet d'exécuter une requête ODBC faisant directement appel à des variables 4D associées :

**C\_TEXTE**(MonTexte)

**C\_ENTIER LONG**(MonEntierLong)

**ODBC LOGIN**("mysql";"root";"")

SQLStmt:="insert into app\_testTable (alpha\_field, longint\_field) VALUES (<<MonTexte>>, <<MonEntierLong>>)"

**Boucle** (vCounter;1;10)

MonTexte:="Texte"+**Chaine**(vCounter)

MonEntierLong:=vCounter

**ODBC EXECUTER**(SQLStmt)

**Fin de boucle**

- ▼ Même exemple que le précédent, mais en utilisant la commande ODBC  
FIXER PARAMETRE :

**C\_TEXTE**(MonTexte)

**C\_ENTIER LONG**(MonEntierLong)

**ODBC LOGIN**("mysql";"root";"")

SQLStmt:="insert into app\_testTable (alpha\_field, longint\_field) VALUES  
(?,?)"

**Boucle** (vCounter;1;10)

MonTexte:="Texte"+**Chaine**(vCounter)

MonEntierLong:=vCounter

**ODBC FIXER PARAMETRE**(MonTexte;ODBC Paramètre entrée)

**ODBC FIXER PARAMETRE**(MonEntierLong;ODBC Paramètre entrée)

**ODBC EXECUTER**(SQLStmt)

**Fin de boucle**

- ▼ Cet exemple permet d'exécuter une requête ODBC utilisant directement des tableaux 4D associés :

**TABLEAU TEXTE**(MonTabTexte;10)

**TABLEAU ENTIER LONG**(MonTabLong;10)

**Boucle** (vCounter;1;**Taille tableau**(MonTabTexte))

MonTabTexte{vCounter}:="Texte"+**Chaine**(vCounter)

MonTabLong{vCounter}::=vCounter

**Fin de boucle**

**ODBC LOGIN**("mysql";"root";"")

SQLStmt:="insert into app\_testTable (alpha\_field, longint\_field) VALUES  
(<<MonTabTexte>>, <<MonTabLong>>)"

**ODBC EXECUTER**(SQLStmt)

- ▼ Cet exemple permet d'exécuter une requête ODBC utilisant directement des champs 4D associés :

**TOUT SELECTIONNER**([Table 2])

**ODBC LOGIN**("mysql";"root";"")

SQLStmt:="insert into app\_testTable (alpha\_field, longint\_field) VALUES  
(<<[Table 2]Champ1>"+>, <<[Table 2]Champ2>)"

**ODBC EXECUTER**(SQLStmt)

## ODBC LIRE DERNIERE ERREUR

ODBC LIRE DERNIERE ERREUR (errCode;errTexte;errODBC;errSQLServer)

Paramètres	Type	Description
errCode	Entier long	← Code de l'erreur
errTexte	Texte	← Texte de l'erreur
errODBC	Texte	← Code d'erreur ODBC
errSQLServer	Texte	← Code d'erreur native SQL server

La commande ODBC LIRE DERNIERE ERREUR retourne des informations relatives à la dernière erreur rencontrée lors de l'exécution d'une commande ODBC. L'erreur peut provenir de l'application 4<sup>e</sup> Dimension, du réseau, de la source ODBC, etc.

Cette commande doit généralement être appelée dans le contexte d'une méthode de gestion des erreurs installée à l'aide de la commande APPELER SUR ERREUR.

- Le paramètre *errCode* retourne le code de l'erreur.
- Le paramètre *errTexte* retourne le libellé de l'erreur.

Les deux derniers paramètres ne sont remplis que si l'erreur provient de la source ODBC. Dans le cas contraire, ils sont retournés vides.

- Le paramètre *errODBC* retourne le code d'erreur ODBC (*SQL state*).
- Le paramètre *errSQLServer* retourne le code de l'erreur native SQL server.

## ODBC IMPORTER

ODBC IMPORTER(tableSource{; projet}{; \*})

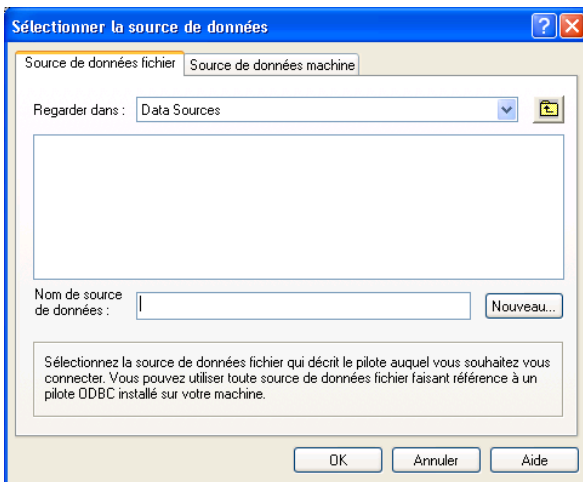
Paramètres	Type	Description
tableSource	Chaîne	→ Nom de la table dans la source de données ODBC
projet	BLOB	→ Contenu du projet d'importation ← Nouveau contenu du projet d'importation (si * est passé)
*	*	→ Affichage de la boîte de dialogue d'importation et mise à jour du projet

La commande ODBC IMPORTER permet d'importer des données depuis la table *tableSource* d'une source ODBC externe. Les paramètres de connexion (nom de la source, utilisateur et mot de passe) sont inclus dans le BLOB *projet*.

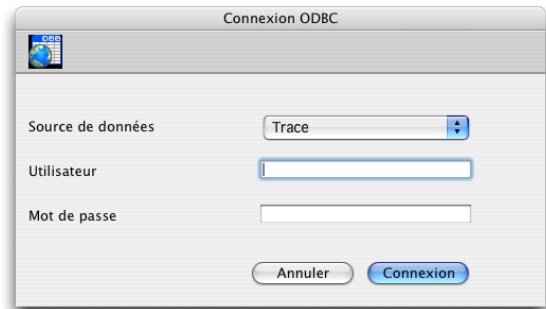
- Notes*
- Le projet contient tous les paramètres de l'importation, notamment la source de données et les tables et champs d'arrivée. Vous définissez ces paramètres dans la boîte de dialogue d'importation ODBC, puis vous pouvez éventuellement les sauvegarder dans un fichier sur disque. Pour plus d'informations, reportez-vous au manuel *Mode Utilisation*.
  - Les projets générés dans la boîte de dialogue d'importation ODBC ne sont pas compatibles avec les commandes et la boîte de dialogue d'importation standard de 4<sup>e</sup> Dimension.

Si vous ne passez pas le paramètre facultatif *projet*, ODBC IMPORTER provoque l'affichage de la boîte de dialogue de sélection de la source de données ODBC :

Windows



MacOS



Une fois la source sélectionnée, la boîte de dialogue d'importation ODBC de 4<sup>e</sup> Dimension apparaît, permettant à l'utilisateur de configurer l'opération. Si l'utilisateur clique sur le bouton **Annuler** dans l'une des deux boîtes de dialogue, l'exécution est stoppée et la variable système *OK* prend la valeur 0.

Si vous passez dans le paramètre *projet* un BLOB contenant un projet d'import ODBC valide, l'importation s'effectue directement, sans intervention de l'utilisateur. Pour cela, il vous suffit de charger un projet préalablement sauvegardé sur disque dans le champ ou la variable BLOB que vous passez dans le paramètre *projet*, à l'aide de la commande DOCUMENT VERS BLOB.

Vous pouvez également utiliser la commande ODBC IMPORTER avec un paramètre *projet* vide et le paramètre optionnel \*, puis stocker le paramètre *projet* dans un champ BLOB (cf. ci-dessous). Cette solution permet, d'une part, de le conserver avec le fichier de données et d'autre part d'éviter la phase de chargement dans un BLOB depuis le disque.

Le paramètre optionnel \*, s'il est spécifié, provoque l'affichage de la boîte de dialogue d'importation de données de 4<sup>e</sup> Dimension avec les paramètres éventuellement définis dans le *projet*. Ce fonctionnement permet d'utiliser un projet prédéfini tout en ayant la possibilité de modifier un ou plusieurs paramètres. En outre, dans ce cas le paramètre *projet* contient, après la fermeture de la boîte de dialogue, les paramètres du "nouveau" projet. Vous pouvez alors le stocker dans un champ BLOB, dans un fichier disque, etc.

Si l'importation se déroule correctement, la variable système OK prend la valeur 1.

**Référence :** ODBC EXPORTER.

## ODBC EXPORTER

ODBC EXPORTER(tableSource{; projet}{; \*})

Paramètres	Type	Description
tableSource	Chaîne	→ Nom de la table dans la source de données ODBC
projet	BLOB	→ Contenu du projet d'exportation ← Nouveau contenu du projet d'exportation (si * est passé)
*	*	→ Affichage de la boîte de dialogue d'exportation et mise à jour du projet

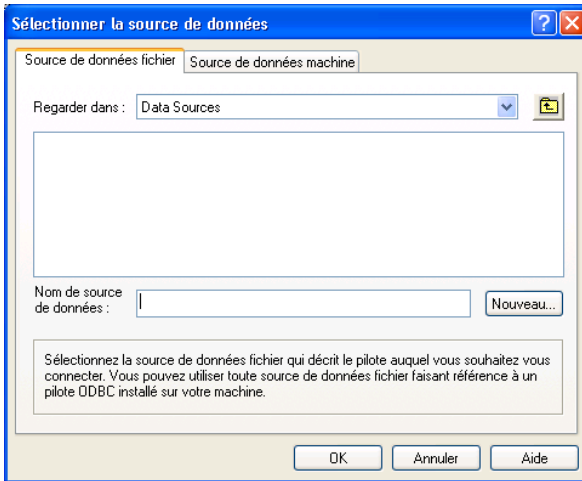
La commande ODBC EXPORTER permet d'exporter des données dans la table *tableSource* d'une source ODBC externe. Les paramètres de connexion (nom de la source, utilisateur et mot de passe) sont inclus dans le BLOB *projet*.

- Notes*
- Le projet contient tous les paramètres de l'exportation, notamment la source de données et les tables et champs exportés. Vous définissez ces paramètres dans la boîte de dialogue d'exportation ODBC, puis vous pouvez éventuellement les sauvegarder dans un fichier sur disque. Pour plus d'informations, reportez-vous au manuel *Mode Utilisation*.
  - Les projets générés dans la boîte de dialogue d'exportation ODBC ne

sont pas compatibles avec les commandes et la boîte de dialogue d'exportation standard de 4<sup>e</sup> Dimension.

Si vous ne passez pas le paramètre facultatif *projet*, ODBC EXPORTER provoque l'affichage de la boîte de dialogue de sélection de la source de données ODBC :

### Windows



### MacOS



Une fois la source sélectionnée, la boîte de dialogue d'exportation ODBC de 4<sup>e</sup> Dimension apparaît, permettant à l'utilisateur de configurer l'opération. Si l'utilisateur clique sur le bouton **Annuler** dans l'une des deux boîtes de dialogue, l'exécution est stoppée et la variable système *OK* prend la valeur 0.

Si vous passez dans le paramètre *projet* un BLOB contenant un projet d'export ODBC valide, l'exportation s'effectue directement, sans intervention de l'utilisateur. Pour cela, il vous suffit de charger un projet préalablement sauvegardé sur disque dans le champ ou la variable BLOB que vous passez dans le paramètre *projet*, à l'aide de la commande DOCUMENT VERS BLOB.

Vous pouvez également utiliser la commande ODBC EXPORTER avec un paramètre *projet* vide et le paramètre optionnel \*, puis stocker le paramètre *projet* dans un champ BLOB (cf. ci-dessous). Cette solution permet, d'une part, de le conserver avec le fichier de données, et d'autre part d'éviter la phase de chargement dans un BLOB depuis le disque.

Le paramètre optionnel \*, s'il est spécifié, provoque l'affichage de la boîte de dialogue de paramétrage d'exportation ODBC avec les paramètres éventuellement définis dans le *projet*. Ce fonctionnement permet d'utiliser un projet prédéfini tout en ayant la possibilité de modifier un ou plusieurs paramètres. En outre, dans ce cas le paramètre *projet* contient, après la fermeture de la boîte de dialogue, les paramètres du "nouveau" projet. Vous pouvez alors le stocker dans un champ BLOB, sur disque, etc.

Si l'exportation se déroule correctement, la variable système OK prend la valeur 1.

**Référence :** ODBC IMPORTER.

## Outils

Le nouveau thème "Outils" regroupe diverses commandes utilitaires :

- **GENERER APPLICATION**, permettant de mettre en place des méthodes automatisées pour la génération d'applications.
- **ENCODER** et **DECODER**, permettant d'encoder et de décoder des BLOBs en base64.
- **CORRECTION ORTHOGRAPHIQUE** et **CHANGER DICTIONNAIRE**, pour gérer la correction orthographique.

### GENERER APPLICATION

GENERER APPLICATION {(nomProjet)}

Paramètres	Type	Description
nomProjet	Chaîne	→ Chemin d'accès complet du projet à utiliser

La commande GENERER APPLICATION lance le processus de génération d'application en prenant en compte les paramètres définis dans le projet d'application courant ou le projet d'application désigné par le paramètre *nomProjet*.

Un projet d'application est un fichier XML contenant tous les paramètres utilisés pour générer une application. La plupart de ces paramètres sont visibles dans la boîte de dialogue du Générateur d'application (à ce sujet, reportez-vous au [paragraphe "Générateur d'applications", page 181](#)). Par défaut, 4<sup>e</sup> Dimension crée pour chaque base de données un projet d'application par défaut nommé

“BuildApp.XML” et placé dans le sous-dossier BuildApp du dossier Preferences de la base.

Si la base n’a pas été compilée ou si le code compilé n’est pas à jour, la commande lance au préalable le processus de compilation. Dans ce cas, la fenêtre du compilateur n’apparaît pas (sauf en cas d’erreur), seule une barre de progression est affichée.

Si vous ne passez pas le paramètre facultatif *nomProjet*, la commande affiche une boîte de dialogue standard d’ouverture de document, vous permettant de désigner un fichier de projet. La variable système *Document* contiendra le chemin d’accès complet du fichier sélectionné. Si vous passez le chemin d’accès et le nom d’un fichier XML de projet d’application valide (extension “.xml”), la commande utilisera les paramètres définis dans le fichier. Pour plus d’informations sur la structure et les clés utilisables dans un fichier XML de projet d’application, reportez-vous au manuel *Clés XML de 4<sup>e</sup> Dimension*.

La variable système *OK* prend la valeur 1 si la commande a été correctement exécutée, sinon elle prend la valeur 0. Dans ce dernier cas, une erreur est générée, que vous pouvez intercepter à l’aide de la commande APPELER SUR ERREUR.

- ▼ Génération de deux applications dans une seule méthode :

```
GENERER APPLICATION("c:\\dossier\\projets\\monprojet1.xml")  
Si (OK=1)  
  GENERER APPLICATION("c:\\dossier\\projets\\monprojet2.xml")  
Fin de si
```

## ENCODER

ENCODER (blob)

Paramètres	Type	Description
blob	BLOB	→ BLOB à encoder en base64 ← BLOB encodé en base64

La commande ENCODER encode le BLOB passé dans le paramètre *blob* en base64. La commande modifie directement le *blob* passé en paramètre.

L’encodage base64 modifie des données codées sur 8 bits afin qu’elles ne conservent plus que 7 bits utiles. Cet encodage est par exemple requis pour la manipulation des BLOBs via le XML.

**Référence :** [DECODER](#)



**DECODER**

## DECODER (blob)

Paramètres	Type	Description
blob	BLOB	→ BLOB encodé en base64 ← BLOB décodé

La commande DECODER permet de décoder le BLOB encodé en base64 passé dans le paramètre *blob*. La commande modifie directement le *blob* passé en paramètre.

La commande n'effectue pas de contrôle sur le contenu du *blob*. Vous devez veiller à ce que les données passées soient effectivement encodées en base64, sinon le résultat obtenu ne sera pas correct.

**Référence :** [ENCODER](#)

**CORRECTION ORTHOGRAPHIQUE**

## CORRECTION ORTHOGRAPHIQUE

Paramètres	Type	Description
		Cette commande ne requiert pas de paramètre

La commande CORRECTION ORTHOGRAPHIQUE déclenche la vérification de l'orthographe du champ ou de la variable ayant le focus dans le formulaire affiché à l'écran. L'objet vérifié doit être de type Alpha ou Texte.

---

*Note* Si vous souhaitez déclencher la correction orthographique à partir d'un bouton dans le formulaire, assurez-vous qu'il ne dispose pas de la propriété "Focusable".

---

La vérification débute par le premier mot du champ ou de la variable. Si un mot inconnu est détecté, la boîte de dialogue de correction apparaît (pour plus d'informations, reportez-vous au [paragraphe "Correction orthographique", page 152](#)). 4<sup>e</sup> Dimension utilise le dictionnaire courant (correspondant à la langue de l'application) sauf si vous avez utilisé la commande [CHANGER DICTIONNAIRE](#).

**Référence :** [CHANGER DICTIONNAIRE](#)

## CHANGER DICTIONNAIRE

### CHANGER DICTIONNAIRE (dictionnaire)

Paramètres	Type	Description
dictionnaire	Entier long	→ Dictionnaire à utiliser pour la correction orthographique

La commande CHANGER DICTIONNAIRE provoque le remplacement du dictionnaire courant par celui spécifié par le paramètre *dictionnaire*. Le dictionnaire courant est utilisé pour la correction orthographique intégrée de 4<sup>e</sup> Dimension (pour plus d'informations sur ce point, reportez-vous au [paragraphe "Correction orthographique", page 152](#)). Par défaut, 4<sup>e</sup> Dimension utilise le dictionnaire correspondant à la langue de l'application. Quatre dictionnaires sont disponibles : français, anglais, allemand et espagnol.

Passez dans le paramètre *dictionnaire* le numéro du dictionnaire à utiliser. Vous pouvez utiliser une des constantes prédéfinies suivantes, placées dans le thème "Dictionnaires" :

Constante	Type	Valeur
Dictionnaire anglais	Entier long	69632
Dictionnaire allemand	Entier long	131584
Dictionnaire espagnol	Entier long	196608
Dictionnaire français	Entier long	262144

Si le dictionnaire est correctement chargé, la variable système *OK* prend la valeur 1, sinon elle prend la valeur 0 et une erreur est retournée.

**Référence :** [CORRECTION ORTHOGRAPHIQUE](#)

# Index

## Symboles / Chiffres

##D	48
& (Editeur de formules)	36
.4BK	206
.4BL	207
.4DA	256
.4DL (fichier d'historique)	220
.bundle	21
.log (Générateur d'applications)	183
.xml (projet d'application)	424
(Editeur de formules)	36
2003 (versions)	15
4D Backup	238
4D Client	
Fichier de connexion	53
4D Customizer Plus	54
4D Engine	181
4D Help	27
4D Help.rsr	27
4D Runtime Classic	181
4D Runtime Interpreted	181
4D Runtime Single User	181
4D Runtime Volume License	181
4D Server	
Commandes d'impression	314
Enregistrer comme service sous MacOS X	26
Numéro de port	68
4D Syntax.rsr	27
4DA	256
6.7.x (versions)	15
6.8.x (versions)	15

## A

Acceptation de saisie (raccourci)	66
Accès à la base durant la sauvegarde	201
Action (Fichier d'historique)	225
Action en cas de double-clic	172
Actions standard	174
Activer la grille magnétique	130
Activer le magnétisme par défaut (Préférences)	61
Activer le suivi d'activité	77
Affichage de la fenêtre d'accueil	65
Affichage de la progression de l'impression	65

Affichage des bibliothèques d'objets	179
Affichage des listes	169
Affichage des objets (Formulaires utilisateurs)	31
Affichage par défaut (Préférences)	61
Afficher enregistrement	173
Afficher l'Explorateur	75
Afficher les en-têtes (List box)	138
AFFICHER NOTIFICATION	316
Afficher ou masquer les objets d'une vue	124
Agrandir (List box)	142
Aide syntaxique	27
Ajout d'un sous-enregistrement (raccourci)	66
AJOUTER A TABLEAU	368
Ajouter enregistrement (Action standard)	173
Ajouter sous-enregistrement (Action standard)	174
Ajustement dynamique	132
Alignement	117
Alignement magnétique	129
ALLER A PAGE	289
ALLER DANS SELECTION	359
Analyser l'historique	223
Annulation de saisie (raccourci)	66
Annuler la sauvegarde après une attente de	212
Application Client-Serveur	185
Applications client/serveur	184
Applications clientes	
Forcer la mise à jour	193
Mise à jour automatique	185
Appliquer la transformation XSL (Export)	38
APPLIQUER TRANSFORMATION XSLT	403
Architecture des applications 4D	17
Architecture des plug-ins	18
Archives	
Compression	213
Segmentation	213
Associer une icône au bouton 3D	160
Aucun	
Mode de sélection	166
Style bouton 3D	157
Avec pop-up menu (Boutons 3D)	161

<b>B</b>	
Backup.XML	237
Badges	118, 125
Barre d'objets	
Editeur de formulaires	118
Formulaires utilisateurs	31
Barre d'outils	71
Editeur de formulaires	115
Editeur de méthodes	110
Barre de défilement horizontale	151
Sous-formulaires	172
Bevel (Style bouton 3D)	159
Bevel arrondi (Style bouton 3D)	159
Bibliothèque d'objets	34, 175, 255
Ouvrir (formulaires utilisateurs)	31
BLOB VERS UTILISATEURS	274
Boîte à outils	77
Énumérations	85
Filtres	87
Formats	87
Groupes	81
Images	83
Infobulles	84
Menus	82
Styles	86
Utilisateurs	79
Boîte de dialogue d'identification	50
Boîte de dialogue de gestion d'accès personnalisée	279
Bordures (option de génération du code 4D)	46
Bouton barre outils (Style bouton 3D)	157
Bouton poussoir (Style bouton 3D)	157
Boutons 3D	155
avec pop-up menu	161
Événements formulaires	303
Style de bouton	157
Boutons image	
Nom/N°	160
Boutons radio groupés par nom	58, 164
BuildApp.XML	182
<b>C</b>	
Cadenas	125
Calcul des références incluses	164
Calcul du cache adaptatif	62
Calcul sur dates (Préférences)	68
Cases à cocher à trois états	154
CGI	366
Chaîne vide (Editeur de formules)	36
Champs	
Editeur de formules	36
Numéros	77
Changement de nom des commandes XML 4D 2003	375
Changement de plan	117
CHANGER COORDONNEES FENETRE	282
CHANGER DICTIONNAIRE	426
CHANGER ELEMENT	267
Changer mot de passe	50
Ne pas afficher le bouton	51
CHANGER PROPRIETES ELEMENT	265, 266
CHANGER PROPRIETES LISTE	263
CHANGER UTILISATEUR	276
CHANGER UTILISATEUR COURANT	276
CHOIX COULEUR	327
CHOIX SAISSABLE	323
CHOIX VISIBLE (List box)	332
Classic	17
Clés XML de 4e Dimension	193
CLI	23
Client serveur exécutable	187
Client-Serveur (Générateur d'applications)	182
Code ASCII du 0 (Préférences)	67
Codes d'erreurs des formulaires utilisateurs	260
Codes d'erreurs Sauvegarde	374
Codes d'erreurs XML	401
Colonnes de list box	141
Command Line Interface	23
Commencer un glisser-déposer (Préférences)	61
Communications	369
Comparaisons (Préférences)	67
Compatibilité	15
états rapides	37
générale	15
plug-ins	18
Supprimer "/" sur les URLs inconnus	59
Compilateur	89
Composants (Explorateur)	95
Compression des archives	213
Compter dans tableau	368
Configuration de l'impression (formulaires utilisateurs)	31
Configuration minimale	16
Configurer la liste de sélection d'objets	112
Connexions persistantes	241

Conserver uniquement N derniers fichiers de sauvegarde . . . . .	209	DERNIERE PAGE . . . . .	289
Constantes de couleur . . . . .	327	Désactiver les warnings . . . . .	89
Construction d'une application client/serveur . . . . .	184	Désenregistrer la base courante . . . . .	26
Construire Application Client-Serveur . . . . .	185	Désenregistrer tous les services 4D Server . . . . .	26
Contents . . . . .	19	Détection et activation des URLs . . . . .	152
Contenu de la sauvegarde . . . . .	204	DIALOGUE . . . . .	317
Contrôles de saisie natifs . . . . .	150	Dictionnaires . . . . .	154
Conversion des fichiers 4D . . . . .	15	Division entière (Editeur de formules) . . . . .	36
Copier dans le Presse-papiers (code 4D généré) . . . . .	47	DOC (Web Services)	
Corbeille		Publication . . . . .	194
Explorateur . . . . .	91	Utilisation . . . . .	195
Récupérer un objet . . . . .	92	Documents système . . . . .	317
Vider la corbeille . . . . .	93	DOM Chercher element XML . . . . .	399
CORRECTION ORTHOGRAPHIQUE . . . . .	153, 425	DOM Créer element XML . . . . .	394
Correction orthographique . . . . .	152	DOM Créer ref XML . . . . .	392
Couleur de fond		DOM ECRIRE ATTRIBUT XML . . . . .	393
alternée (Colonne de List box) . . . . .	144	DOM ECRIRE NOM ELEMENT XML . . . . .	396
Colonne de List box . . . . .	144	DOM ECRIRE OPTION XML . . . . .	391
List box . . . . .	140	DOM ECRIRE VALEUR ELEMENT XML . . . . .	397
Couleur de fond alternée (List box) . . . . .	140	DOM EXPORTER VERS FICHER . . . . .	398
Couleur lignes horizontales (List box) . . . . .	139	DOM EXPORTER VERS VARIABLE . . . . .	399
Couleur lignes verticales (List box) . . . . .	139	DOM LIRE VALEUR ELEMENT XML . . . . .	400
Couleurs		DOM SUPPRIMER ELEMENT XML . . . . .	398
Lignes des taquets . . . . .	61	Dossier 4D Client . . . . .	186
polices (List box) . . . . .	138	Dossier 4D Server . . . . .	186
tables et champs . . . . .	77	Dossier Extras . . . . .	180, 258
Création et gestion des valeurs de list box . . . . .	329	Dossier Macros . . . . .	109
Créer fenetre . . . . .	288	Dossier PlugIns . . . . .	19
Créer fenetre formulaire . . . . .	288	Dossiers (Explorateur) . . . . .	96
CREER FORMULAIRE UTILISATEUR . . . . .	258	Ajouter un objet lors de sa création . . . . .	103
Customizer Plus, voir 4D Customier Plus		Dossiers par défaut . . . . .	99
		Groupes . . . . .	105
		Informations . . . . .	105
		Renommer . . . . .	100
		Supprimer . . . . .	101
		Double-clic sur ligne . . . . .	173
		Double-clic sur ligne vide . . . . .	173
<b>D</b>		<b>E</b>	
Date et heure		Ecart type . . . . .	48
Fichier d'historique . . . . .	225	Echec de la sauvegarde . . . . .	202
Sauvegarde . . . . .	207	ECRIRE ACCES PLUGIN . . . . .	271
Date vierge (Editeur de formules) . . . . .	36	Ecrire proprietes utilisateur . . . . .	278
Décalage du fond (Style bouton 3D) . . . . .	157	Ecriture du cache . . . . .	76
Décalage icône (Boutons 3D) . . . . .	163	EDITER ELEMENT . . . . .	312
Déclencher une sauvegarde . . . . .	199	List box . . . . .	332
Découpage de l'image (Boutons 3D) . . . . .	160	Editer formule . . . . .	253
DEFILER LIGNES . . . . .	309	Editer un élément de liste . . . . .	76
Délai avant déconnexion . . . . .	243		
Déplacement . . . . .	116		
DEPLACER OBJET . . . . .	132		
Déplacer vers vue . . . . .	123		
Dernier objet . . . . .	311, 332		

Editeur d'états rapides	45	Evénements formulaire liés aux enregistrements en liste	306
Editeur de formulaires	111	Evénements formulaire liés aux list box	304
Editeur de formulaires utilisateurs	29	Evénements formulaire liés aux listes hiérarchiques	307
Editeur de formules	35	Evénements formulaire liés aux survols	302
Editeur de méthodes	106	Exclusion (List box)	143
Effacer la sauvegarde la plus ancienne	211	Exécution de la sauvegarde	201
Effacer tous les modèles personnalisés	129	Explorateur	90
Préférences	60	Affichage	75
Element parent	268	Composants	95
Element selectionne	268	Dossiers	96
Elements selectionnes	268	Page Corbeille	91
Emplacement de l'icône (List Box)	146	Explorateur d'exécution	77
Emplacements pour les plug-ins	18	Exponentiation (Editeur de formules)	36
ENCODER	424, 425	Exportation ODBC	
Enregistrements		Enregistrer préférences	45
Exportation ODBC	44	Exporter vers la source ODBC	43
Importation ODBC	42	Extras	180, 258
Enregistrer comme service sous MacOS X	26		
Enregistrer l'accès à la base (4D Server)	52		
En-têtes de list box	145		
Enumération (List box)	141		
Enumérations (Boîte à outils)	85		
Environnement 4D	348		
Environnement système	352		
ENVOYER DONNEES HTTP	363		
Erreur -10002	65		
et suivant (Préférences)	67		
Etats rapides			
Compatibilité	37		
Ecart type	48		
Générer le code	45		
Evénements formulaire	301		
List box	140, 332		
Sur après tri	306		
Sur avant saisie	304		
Sur chargement ligne	306		
Sur clic	148		
Sur clic entête	305		
Sur clic flèche	303		
Sur clic long	303		
Sur contracter	307		
Sur début survol	302		
Sur déplacement colonne	305		
Sur déplacement ligne	305		
Sur déployer	307		
Sur fin survol	302		
Sur nouvelle sélection	305, 306, 307		
Sur redimensionnement colonne	305		
Sur survol	302		
		<b>F</b>	
		Fenêtre d'accueil	
		Affichage (Préférences)	65
		Fenêtre de compilation	89
		Fenêtre feuille	287
		Fenetre formulaire courant	280
		Fenêtre pop up	286
		Fenêtres	280
		Bords pousseurs	283
		Fermer l'historique	223
		Fichier d'aide syntaxique	27
		Fichier d'historique	217
		Création	220
		Fermer	223
		Générateur d'applications	183
		Intégrer le dernier si la base est incomplète	230
		Fichier de connexion 4D Client	53
		Fichier de données (Sauvegarde)	204
		Fichier de structure (Sauvegarde)	204
		Fichier de structure utilisateur (Sauvegarde)	204
		Fichier XML paramètres de sauvegarde	236
		Fichiers de sauvegarde	203
		Compression	213
		Emplacement	205
		Intégrité	214
		Nom	206
		Segmentation	213
		Fichiers joints (Sauvegarde)	205

- Filtre de saisie (List box) ..... 143
- Filtres ..... 75
- Filtres (Boîte à outils) ..... 87
- FIXER COULEUR GRILLE LISTBOX ..... 347
- FIXER COULEURS RVB ..... 326
- FIXER EXECUTABLE CGI ..... 366
- FIXER HAUTEUR LIGNES LISTBOX ..... 342
- FIXER LARGEUR COLONNE LISTBOX ..... 337
- FIXER LIEN CHAMP ..... 299
- FIXER LIENS AUTOMATIQUES ..... 295
- FIXER METHODES AUTORISEES ..... 251
- FIXER OPTION IMPRESSION ..... 315
- FIXER OPTION WEB SERVICE ..... 406
- FIXER PARAMETRE XSLT ..... 404
- FIXER REDIMENSIONNEMENT FORMULAIRE HORIZONTAL ..... 291
- FIXER REDIMENSIONNEMENT FORMULAIRE VERTICAL ..... 292
- FIXER TAILLE FORMULAIRE ..... 289
- FIXER TITRES CHAMPS ..... 310
- FIXER TITRES TABLES ..... 310
- FIXER VARIABLE ENVIRONNEMENT ..... 352
- Fond
- Couleur (List box) ..... 138
- Format Mach-O ..... 17
- Format type (List box) ..... 144
- Formats (Boîte à outils) ..... 87
- FORMULAIRE ENTREE ..... 294
- FORMULAIRE SORTIE ..... 295
- Formulaires ..... 289
- Modifiable par l'utilisateur ..... 254
  - Paramètres d'impression ..... 120
- Formulaires utilisateurs ..... 29, 254
- Accès aux bibliothèques d'objets ..... 34
  - Bibliothèque d'objets ..... 255
  - Codes d'erreurs ..... 260
  - Menu contextuel ..... 34
  - Propriétés verrouillées ..... 32
- Formule-fichier ..... 171
- Formules ..... 251
- Editeur ..... 35
- G**
- Générateur d'applications ..... 181
- Gestion des plug-ins ..... 189
  - Numéros de série ..... 190
- GENERER APPLICATION ..... 423
- Générer le code ..... 47
- Générer le code 4D (Etats rapides) ..... 45
- Gestion de la saisie ..... 312
- Gestion des plug-ins (Générateur d'applications) ... 189
- Gestion des sauvegardes ..... 198
- Gestion du fichier d'historique ..... 75
- Glisser-déposer ..... 151
- Grille magnétique ..... 129
- Groupement/Dégroupement ..... 117
- Grouper les boutons radio ..... 164
- Groupes (Boîte à outils) ..... 81
- Groupes et dossiers ..... 105
- H**
- Hauteur des lignes (List box) ..... 139
- Heure vierge (Editeur de formules) ..... 36
- Historique
- Historique obligatoire ..... 221
  - Revenir en arrière ..... 234
- I**
- Icône (List Box) ..... 146
- Icônes associées aux boutons 3D ..... 160
- Identification de l'utilisateur ..... 50
- Images (Boîte à outils) ..... 83
- Importation et exportation ..... 37
- ODBC ..... 38
- Importation ODBC
- Enregistrer préférences ..... 42
- Importer de la source ODBC ..... 40
- Impression
- Formulaires ..... 120
  - Interface de plate-forme ..... 149
- Impressions ..... 314
- IMPRIMER ENREGISTREMENT ..... 315
- IMPRIMER ETIQUETTES ..... 315
- Imprimer ligne ..... 315
- IMPRIMER SELECTION ..... 59, 315
- Indexation durant la sauvegarde ..... 211
- Infobulles
- Boîte à outils ..... 84
  - List Box ..... 146
- Information de publication (Préférences) ..... 69
- INFORMATION ELEMENT ..... 267
- Informations sur la dernière sauvegarde ..... 207
- INSERER COLONNE LISTBOX ..... 334
- INSERER LIGNE LISTBOX ..... 340

Insérer un lien vers le fichier XSL .....	38	LIRE ENREGISTREMENTS MARQUES .....	355
Inspecteur .....	76	LIRE ERREUR XSLT .....	405
Installation des plug-ins .....	18	Lire formatage .....	321
Intégration manuelle de l'historique .....	234	Lire hauteur lignes listbox .....	342
Intégrer le dernier historique si la base est incomplète .....	230	Lire information listbox .....	345
Intégrité des fichiers de sauvegarde .....	214	LIRE INFORMATION RESTITUTION .....	372
Interface .....	71	LIRE INFORMATION SAUVEGARDE .....	371
Interface de plate-forme .....	148	Lire largeur colonne listbox .....	338
Compatibilité .....	149	LIRE LIEN CHAMP .....	296
Interface utilisateur .....	308	LIRE LIENS AUTOMATIQUES .....	301
Introduction .....	13	LIRE LISTE PLUGIN .....	273
Invisible (List box) .....	144	LIRE METHODES AUTORISEES .....	252
Invisible par défaut .....	165	Lire nombre colonnes listbox .....	336
		Lire nombre lignes listbox .....	341
		LIRE NOMS COLONNES LISTBOX .....	344
		LIRE OBJETS FORMULAIRE .....	281, 292
		LIRE OPTION IMPRESSION .....	315
		LIRE PROPRIETES FORMULAIRE .....	289
		LIRE PROPRIETES LISTE .....	264
		LIRE PROPRIETES UTILISATEUR .....	278
		LIRE TITRES CHAMPS .....	309
		LIRE TITRES TABLES .....	308
		Lire utilisateur par défaut .....	275
		List box .....	133
		CHOIX COULEUR .....	327
		Commandes .....	328
		Événements formulaire .....	304
		Principes de programmation .....	329
		Propriétés des colonnes .....	141
		Propriétés des en-têtes .....	145
		Propriétés spécifiques .....	137
		Utilisation .....	135
		Liste des commandes (Editeur de formules) .....	36
		Liste des propriétés .....	112
		Configurer la liste de sélection d'objets .....	112
		Formulaires utilisateurs .....	31
		Liste des tables .....	75
		Liste des vues .....	121
		LISTE FORMULAIRES UTILISATEURS .....	259
		Listes d'enregistrements .....	165
		Affichage des lignes vides .....	170
		Affichage des listes .....	169
		Modes de sélection .....	166
		Saisie en liste .....	168
		Listes hiérarchiques .....	260
		Événements formulaire .....	307
		Localhost .....	247
<b>J</b>			
Journal des sauvegardes .....	236		
<b>K</b>			
Keep-alive .....	241		
<b>L</b>			
L'utilisateur peut changer son mot de passe .....	52		
Lancement du serveur Web sous MacOS X .....	245		
Lancer la base après la restitution automatique .....	230		
LANCER PROCESS EXTERNE .....	353		
Largeur (List box) .....	142, 146		
Largeur maxi (List box) .....	142		
Largeur mini (List box) .....	142		
Les champs ne sont pas saisissables dans les dialogues (Préférences) .....	58		
Licence 4D SOAP .....	195		
Licence disponible .....	195		
Licence SOAP 4D Client .....	195		
Licences (Générateur d'applications) .....	182, 191		
Lié (pop up menu) .....	162		
Liens .....	295		
LIENS AUTOMATIQUES .....	295		
Ligne unique (mode de sélection) .....	167		
Lignes déplaçables (List box) .....	140		
Lignes horizontales (List box) .....	139		
Lignes verticales (List box) .....	139		
Lignes vides (affichage) .....	170		
Lire acces plugin .....	272		
LIRE CORPS HTTP .....	362		
LIRE CORRESPONDANCE PORT SERIE .....	369		



**M**

Mac4DX	20
Mach-O	17
MacOS 9	17
MacOS X	
Lancement du serveur Web	245
Macro-commandes	
Method	106
Macros (nouveau dossier)	109
Manuel	14
Marge horizontale / Marge verticale	163
MARQUER ENREGISTREMENTS	357
Mémoire	16
Mémoire cache	
Mode "forcé"	62
Mode adaptatif	62
Mémoire physique à réserver	63
Mémoire principale	61
Menu contextuel (formulaires utilisateurs)	34
Menus	72
Boîte à outils	82
Messages	316
method (macro-commande)	106
Méthode base Sur arrêt sauvegarde	373
Méthode base Sur démarrage sauvegarde	373
Méthodes	
Accès dans l'éditeur de formules	37
objet (List box)	331
Prévisualisation	95
Mise à jour automatique de l'application cliente	185
Mode adaptatif	62
Mode Structure	49
Mode Utilisation	29
Modèles	
Effacer	129
Modes de sélection (listes)	166
Modifiable par l'utilisateur	29, 130
Formulaire	254
Modification du mot de passe	50
Modifier enregistrement (Action standard)	173
MODIFIER FORMULAIRE	256
MODIFIER SELECTION	358
Modifier sous-enregistrement (Action standard)	174
Module de sauvegarde intégré	197
Monoposte (Générateur d'applications)	182
MONTRER BARRES DEFILEMENT LISTBOX	348
MONTRER GRILLE LISTBOX	346

Mots de passe	50
Enregistrer l'accès à la base	54
Multilignes (mode de sélection)	167

**N**

Ne rien faire	173
Niveau supérieur (Dossiers)	103
Nombase[0000]	206
Nomhistorique[0000]	207
Nom de l'objet (List Box)	145
Nom de la variable (List Box)	145
Nom du fichier de sauvegarde	208
Nom du fichier de sauvegarde d'historique	208
Nom/ID de l'image (Boutons 3D)	160
Nom/ID du fond (Boutons 3D)	163
Nom/N° (List Box)	146
Nomarchive-Restore	233
Nombre de colonnes (List box)	137
Nombre de colonnes fixes (List box)	137
Nombre de requêtes par connexion	243
Nombre elements	270
NomProjet.log (Générateur d'applications)	183
Notation XPath	389
NUMERO COLONNE LISTBOX DEPLACEE	138, 339
Numero de ligne affichee	356
Numéro de port (Préférences)	68
Numéro de port HTTPS	243
NUMERO LIGNE LISTBOX DEPLACEE	343
Numéros de série (Générateur d'applications)	190
Numéros des tables et des champs	77

**O**

Objet focus	311
List box	332
Objets	
Alignement magnétique	129
Type booléen	163
Objets actifs	172
Objets dans le corps des formulaires	169
Objets de formulaires	133
Objets groupés (Sélectionner)	129
Obligation (List box)	143
ODBC (Importation et exportation)	38
ODBC ANNULER CHARGEMENT	415
ODBC CHARGER ENREGISTREMENT	415
ODBC EXECUTER	412
ODBC EXPORTER	421

ODBC Fin de selection	414	Polices	
ODBC FIXER OPTION	411	Tableau de couleur (List box)	138
ODBC FIXER PARAMETRE	416	Port HTTPS	243
ODBC IMPORTER	419	Port TCP 4D Server	68
ODBC LIRE DERNIERE ERREUR	419	Position de l'image (Boutons 3D)	161
ODBC LIRE OPTION	412	Position titre/image (Boutons 3D)	161
ODBC LOGIN	409	Pourcentage de mémoire disponible utilisé pour le cache	63
ODBC LOGOUT	410	Pousseurs	
Office XP (Style bouton 3D)	159	Bords des fenêtres	283
Opérateurs (Éditeur de formules)	36	Séparateurs	147
Options d'exportation ODBC	44	Préférences	54
Options d'importation ODBC	42	Activer le magnétisme par défaut	61
Options dans les requêtes Web Services	196	Affichage par défaut	61
Ordre de saisie	116	Boutons radio groupés par nom	58
Organiser les dossiers	102	Commencer un glisser-déposer	61
Orthographe	152	Compatibilité	58
Outils	423	Effacer tous les modèles personnalisés	60
Ouverture de la bibliothèque (formulaires utilisateurs)	31	L'utilisateur peut changer son mot de passe	52
OUVRIR PREFERENCES 4D	348	Les champs ne sont pas saisissables dans les dialogues	58
		Mémoire cache	62
		Numéro de port	68
		Provoquer un défilement automatique	60
		Publication	69
		Raccourcis clavier	66
		Recharger le formulaire pour chaque enregistrement durant un IMPRIMER SELECTION	59
		Script Manager	67
		Supprimer "/" sur les URLs inconnus	59
		Tableau de comparaison	55
		Taille du pas de déplacement	60
		WEDD	66
		PREMIERE PAGE	289
		Prise en charge avancée du mode DOC	195
		Prise en charge des CGI	366
		Prise en charge des URLs inconnus	244
		Prochaine sauvegarde prévue	208
		Progiciels (packages)	19
		Progression de l'impression (Préférences)	65
		Projet d'application	424
		Propriétés	
		Colonnes de List box	141
		En-têtes de List box	145
		Méthodes	96
		Objets	321
		Texte (option de génération du code 4D)	46
		verrouillées (formulaires utilisateurs)	32
<b>P</b>			
Packages	19		
Page formulaire courante	289		
PAGE PRECEDENTE	289		
PAGE SUIVANTE	289		
Pages formulaire	117, 289		
Palette des propriétés	112		
Paquets (packages)	19		
Paramètres d'impression des formulaires	120		
Pas de badges	126		
Périodicité (Sauvegarde)	214		
Permettre la mise à jour automatique de l'application cliente	192		
Personnalisé (Style bouton 3D)	158		
Personnaliser les dossiers 4D Server et/ou 4D Client	186		
Petit carré système (Style bouton 3D)	158		
Plage de valeurs (List box)	143		
Plug-ins			
Architecture	21		
Compatibilité	18		
Emplacements	18		
Générateur d'applications	182, 189		
Installation	18		
Priorité	21		
PlugIns (dossier)	19, 20		

- Provoquer un défilement automatique (Préférences)  
60
- Publication (Préférences) ..... 69
- Publication en mode DOC (Web Services) ..... 194
- Publier ..... 69
- Q**
- qr cmd écart type ..... 48
- qr écart type ..... 48
- QR ETAT ..... 315
- QR EXECUTER COMMANDE ..... 48
- QR FIXER DONNEES TOTAUX ..... 48
- QR LIRE DONNEES TOTAUX ..... 48
- QR Lire statut commande ..... 48
- QR Opérateurs ..... 48
- Quadrillage (List box) ..... 139
- Quitter la liste ..... 359
- R**
- Raccourcis clavier  
Préférences ..... 66  
standard ..... 75
- Recharger le formulaire pour chaque enregistrement  
durant un IMPRIMER SELECTION (Préférences)  
..... 59
- Recherches dans la base ..... 71
- Reconnexion automatique des clients ..... 64
- Récupération des données ..... 227
- Redimensionnable (List box) ..... 142, 143
- Réessayer à la nouvelle date et heure programmée .  
212
- Réessayer dans N heure(s) ou minute(s) ..... 212
- Remonter en arrière ..... 234
- Réorganisation des menus ..... 72
- Réparations automatiques ..... 228
- Répartition ..... 117
- Requêtes ODBC ..... 416
- Réseau (Explorateur d'exécution) ..... 77
- Résolution écran ..... 16
- Ressources TRIC (Préférences) ..... 68
- RESTITUER ..... 371
- Restituer la dernière sauvegarde si la base est endom-  
magée ..... 229
- Restitution manuelle d'une sauvegarde ..... 231
- Revenir en arrière ..... 234, 235
- Rond (Style bouton 3D) ..... 158
- S**
- Saisie ..... 317
- Saisie en liste ..... 75, 168
- Saisissable (List box) ..... 143
- Sauvegarde ..... 370  
automatique ..... 217  
automatique périodique ..... 200  
Codes d'erreurs ..... 374  
En cas d'échec ..... 202  
Fichier XML des paramètres ..... 236  
Gestion ..... 198  
Gestion des indexations ..... 211  
Gestion des transactions ..... 211  
Incidents et diagnostic ..... 227  
Journal ..... 236  
manuelle ..... 199  
Méthode base Sur arrêt sauvegarde ..... 373  
Méthode base Sur démarrage sauvegarde . . . 373  
périodiques ..... 214  
programmée ..... 201  
Restitution manuelle ..... 231  
Revenir en arrière ..... 234
- Sauvegarde des paramètres d'accès à la base ..... 52
- Sauvegarde par défaut ..... 216
- SAUEGARDER ..... 370
- Sauvegarder ..... 198
- Sauvegarder uniquement si le fichier de données a été  
modifié ..... 210
- SAX ..... 375
- SAX AJOUTER CDATA XML ..... 381
- SAX AJOUTER COMMENTAIRE XML ..... 378
- SAX AJOUTER DOCTYPE XML ..... 377
- SAX AJOUTER INSTRUCTION DE TRAITEMENT ..... 383
- SAX AJOUTER VALEUR ELEMENT XML ..... 381
- SAX ECRIRE OPTIONS XML ..... 376
- SAX FERMER ELEMENT XML ..... 380
- SAX LIRE CDATA XML ..... 388
- SAX LIRE COMMENTAIRE XML ..... 385
- SAX LIRE ELEMENT XML ..... 386
- SAX LIRE ENTITE XML ..... 389
- SAX LIRE INSTRUCTION DE TRAITEMENT XML ..  
386
- SAX Lire noeud XML ..... 383
- SAX LIRE VALEUR ELEMENT XML ..... 387
- SAX LIRE VALEURS DOCUMENT XML ..... 385
- SAX OUVRIR ELEMENT XML ..... 378
- SAX OUVRIR ELEMENT XML TABLEAUX ..... 379
- Script Manager (Préférences) ..... 67

Segmentation des sauvegardes et des sauvegardes d'historique .....	213	SUPPRIMER LIGNE LISTBOX .....	341
Segmenter les archives .....	213	Sur affichage corps .....	306
Sélection multiple (List box) .....	138	Sur après tri .....	306
Sélectionner document .....	317	Sur arrêt sauvegarde (méthode base) .....	373
SELECTIONNER ELEMENT .....	261	Sur avant saisie .....	304
SELECTIONNER ELEMENT PAR REFERENCE .....	262	Sur chargement ligne .....	306
SELECTIONNER ELEMENTS PAR POSITION .....	261	Sur clic .....	148
SELECTIONNER ELEMENTS PAR REFERENCE .....	262	Sur clic entête .....	305
SELECTIONNER LIGNE LISTBOX .....	339	Sur clic flèche .....	303
Sélectionner objets de même type .....	124	Sur clic long .....	303
Sélections .....	355	Sur contracter .....	307
Self (List box) .....	332	Sur début survol .....	302
Séparateurs .....	146	Sur démarrage sauvegarde (méthode base) .....	373
Evenement Sur clic .....	148	Sur déplacement colonne .....	305
pousseurs .....	147	Sur déplacement ligne .....	305
Séparé (pop up menu) .....	162	Sur déployer .....	307
Séquenceur .....	77	Sur entête .....	306
Serveur Web .....	241	Sur fin survol .....	302
Commandes .....	360	Sur impression corps .....	59
Connexions persistantes .....	242	Sur nouvelle sélection .....	305, 306, 307
Tester .....	246	Sur redimensionnement colonne .....	305
URLs inconnus .....	59	Sur survol .....	302
Service (Enregistrer 4D Server comme) .....	26	Système (Interface de plate-forme) .....	148
Sheet window .....	287	Système d'exploitation .....	16
Si enreg. non sélectionné .....	171		
Si enreg. sélectionné .....	170	<b>T</b>	
Si la sauvegarde échoue .....	212	Table .....	289
Signature WEDD .....	67	Fichier d'historique .....	225
Source de données		Numéros .....	77
Exportation ODBC .....	44	Table de configuration Accepter/Refuser (Préférences)	69
externes (thème) .....	408	Table SQL .....	42
Importation ODBC .....	42	Tableau de couleurs de fond (List box) .....	138
List box .....	141	Tableau de couleurs de police (List box) .....	138
ODBC .....	38	Tableau de styles (List box) .....	138
Source de l'image (Boutons 3D) .....	160	Tableaux .....	368
Source du fond (Boutons 3D) .....	163	Taille du pas de déplacement (Préférences) .....	60
Sous-formulaires .....	172	Taille du segment .....	213
Action en cas de double-clic .....	172	Taille maximale (Mémoire cache) .....	63
Barre de défilement horizontale .....	172	Taille minimale (Mémoire cache) .....	63
SQL (importation et exportation) .....	39	Taquets	
Statut (Sauvegarde) .....	207	Couleurs des lignes .....	61
Stratégie de sauvegarde .....	216	Taux d'entrelacement .....	214
Style de bouton (Boutons 3D) .....	157	Taux de compression .....	213
Style de la bordure (List box) .....	140	Taux de redondance .....	214
Styles (Boîte à outils) .....	86	Téléchargement des plug-ins en client-serveur ..	22
Supprimer "/" sur les URLs inconnus .....	59	Terminal (MacOS X) .....	23
SUPPRIMER COLONNE LISTBOX .....	335	Tester le serveur Web .....	246
SUPPRIMER FORMULAIRE UTILISATEUR .....	259		

- Texte
- Colonne de list box . . . . . 144
  - En-tête de List Box . . . . . 146
  - List box . . . . . 140
  - Thème Communications . . . . . 369
  - Thème Documents système . . . . . 317
  - Thème Environnement 4D . . . . . 348
  - Thème Environnement système . . . . . 352
  - Thème Fenêtres . . . . . 280
  - Thème Formulaire . . . . . 289
  - Thème Formulaire utilisateurs . . . . . 254
  - Thème Formules . . . . . 251
  - Thème Gestion de la saisie . . . . . 312
  - Thème Impressions . . . . . 314
  - Thème Interface utilisateur . . . . . 308
  - Thème Liens . . . . . 295
  - Thème List box . . . . . 328
  - Thème Listes hiérarchiques . . . . . 260
  - Thème Messages . . . . . 316
  - Thème Outils . . . . . 423
  - Thème Propriétés des objets . . . . . 321
  - Thème Saisie . . . . . 317
  - Thème Sauvegarde . . . . . 370
  - Thème Sélections . . . . . 355
  - Thème Serveur Web . . . . . 360
  - Thème Source de données externes . . . . . 408
  - Thème Tableaux . . . . . 368
  - Thème Utilisateurs et groupes . . . . . 271
  - Thème Web Services (Client) . . . . . 406
  - Thème XML . . . . . 374
- Titre
- Boutons 3D . . . . . 156
  - List Box . . . . . 145
  - Toujours attendre la fin des opérations . . . . . 211
  - Toujours visible . . . . . 170
  - Tout sélect. dans... . . . . 124
  - TRAITER BALISES HTML . . . . . 360
  - Transactions durant la sauvegarde . . . . . 211
  - Transformation XSL . . . . . 401
    - Exportation . . . . . 38
  - Triable (List box) . . . . . 140
  - TRIC (Préférences) . . . . . 67
  - TRIER COLONNES LISTBOX . . . . . 336
  - Trier le contenu de la fenêtre de recherche . . . . . 72
  - Type booléen pour les variables d'objets . . . . . 163
- U**
- URLs (Détection et activation) . . . . . 152
  - Utilisateurs
    - Boîte à outils . . . . . 79
    - Changer mot de passe . . . . . 50
    - Fichier d'historique . . . . . 225
  - Utilisateurs et groupes . . . . . 271
  - UTILISATEURS VERS BLOB . . . . . 273
  - Utiliser comme modèle . . . . . 128
  - Utiliser la reconnexion automatique des clients . . . . . 64
  - Utiliser le fichier d'historique . . . . . 220
- V**
- Valeurs des champs (Fichier d'historique) . . . . . 226
  - Valider mot de passe . . . . . 277
  - Variable d'en-tête et gestion des tris . . . . . 332
  - Variable d'objet et gestion des sélections . . . . . 333
  - Variables (Type booléen) . . . . . 163
  - Vérifier fichier d'historique . . . . . 224
  - Verrouillage
    - Liste des propriétés . . . . . 115
    - objets d'une vue . . . . . 124
    - utilisateur . . . . . 116
  - Version courante . . . . . 185
  - versions 6.7.x . . . . . 15
  - versions 6.8.x . . . . . 15
  - Vider la corbeille . . . . . 93
  - Visibilité des objets . . . . . 170
  - Visibilité des palettes (option de génération du code 4D) . . . . . 46
  - Visible (listes) . . . . . 170
  - VISUALISER SELECTION . . . . . 358
  - Vues . . . . . 118, 121
    - Afficher ou masquer . . . . . 124
    - Placer un objet . . . . . 122
    - Renommer . . . . . 123
    - Sélectionner tout . . . . . 123
    - Tout sélectionner dans . . . . . 124
    - Verrouiller objets . . . . . 124
- W**
- Warnings (désactiver) . . . . . 89
  - Web
    - Connexions persistantes . . . . . 242
    - Tester le serveur Web . . . . . 246
  - Web Services . . . . . 194

## Index

---

Web Services (Client) .....	406
WEDD (Préférences) .....	66
Win4DX .....	20

## X

XML .....	374
Codes d'erreurs .....	401
Notation XPath .....	389
XPath .....	389
XPath .....	389
XSL .....	401
Exportation .....	38
XSLT .....	401

## Z

Zoom .....	116, 120
------------	----------