



4D v11 SQL Release 1 (11.1)

ADDENDUM

Bienvenue dans la *release* 1 de 4D v11 SQL. Ce document présente les nouveautés et modifications proposées dans cette nouvelle version de 4D.

Analyse des recherches

4D propose trois nouvelles commandes permettant d'analyser avec précision le déroulement des recherches (requêtes) effectuées sur les données. Ces commandes sont destinées aux développeurs souhaitant optimiser leurs applications.

Ces trois nouvelles commandes sont placées dans le thème "Recherches et tris".

DECRIRE EXECUTION RECHERCHE

DECRIRE EXECUTION RECHERCHE (statut)

Paramètres	Type	Description
statut	Booléen	→ Vrai=Enregistrer la description des requêtes Faux=Stopper l'enregistrement

La commande **DECRIRE EXECUTION RECHERCHE** permet d'activer ou d'inactiver le mode d'analyse de l'exécution des recherches pour le process courant. La commande tient compte indifféremment des recherches effectuées via le langage 4D et des requêtes SQL.

L'appel de la commande avec le paramètre *statut* à *Vrai* active le mode d'analyse des recherches. Dans ce mode, le moteur de 4D enregistrera en interne deux séries d'informations spécifiques lors de chaque requête effectuée par la suite sur les données :

- la description détaillée de la recherche juste avant son exécution, c'est-à-dire la recherche prévue (le plan de recherche),
- la description détaillée de la recherche telle qu'elle a réellement été exécutée (le chemin de recherche).

Les informations enregistrées incluent le type de recherche (indexée, séquentielle), le nombre d'enregistrements trouvés et le temps nécessaire à l'exécution de chaque critère de recherche.

Vous pouvez ensuite lire ces informations à l'aide des nouvelles commandes [Lire dernier plan recherche](#) et [Lire dernier chemin recherche](#).

En général, la description du plan d'une recherche et celle de son chemin sont identiques, mais elles peuvent toutefois différer car 4D peut mettre en oeuvre des optimisations dynamiques au cours de l'exécution de la recherche, dans le but d'améliorer les performances. Par exemple, une recherche indexée peut être convertie dynamiquement en recherche séquentielle si le moteur de 4D estime qu'elle sera plus rapide — c'est parfois le cas notamment lorsque le nombre d'enregistrements parmi lesquels effectuer la recherche est faible.

Passez Faux dans le paramètre *statut* lorsque vous n'avez plus besoin d'analyser les recherches. Le mode d'analyse de l'exécution des recherches peut ralentir l'application.

- ▼ L'exemple suivant illustre le type d'information obtenue via ces commandes en cas de requête SQL :

C_TEXTE(\$vResultPlan;\$vResultPath)

TABLEAU TEXTE(aTitles;0)

TABLEAU TEXTE(aDirectors;0)

DECRIRE EXECUTION RECHERCHE(Vrai) `mode analyse

Debut SQL

```
SELECT ACTORS.FirstName, CITIES.City_Name
FROM ACTORS, CITIES
WHERE ACTORS.Birth_City_ID=CITIES.City_ID
ORDER BY 1
INTO :aTitles, :aDirectors;
```

Fin SQL

\$vResultPlan:=[Lire dernier plan recherche](#)(Description format Texte)

\$vResultPath:=[Lire dernier chemin recherche](#)(Description format Texte)

DECRIRE EXECUTION RECHERCHE(Faux) `Fin du mode analyse

A l'issue de l'exécution de ce code, *\$vResultPlan* et *\$vResultPath* contiennent les descriptions des recherches effectuées, par exemple :

- *\$vResultPlan* :

[Join] : ACTORS.Birth_City_ID = CITIES.City_ID

■ *\$vResultPath* :

And

[Merge] : ACTORS with CITIES

[Join] : ACTORS.Birth_City_ID = CITIES.City_ID (1227 records found in 13 ms)

--> 1227 records found in 13 ms

--> 1227 records found in 14 ms

Si la constante Description format XML est passée à la commande Lire dernier chemin recherche, *\$vResultPath* contient la description de la recherche exprimée en XML :

```
<QueryExecution>
  <steps description="And" time="0" recordsfound="1227">
    <steps description="[Merge] : ACTORS with CITIES" time="13"
recordsfound="1227">
      <steps description="[Join] : ACTORS.Birth_City_ID =
CITIES.City_ID" time="13" recordsfound="1227"/>
    </steps>
  </steps>
</QueryExecution>
```

Lire dernier plan recherche

Lire dernier plan recherche (formatDesc) → Chaîne

Paramètres	Type	Description
formatDesc	Entier long →	Format de description : Texte ou XML
Résultat	Chaîne ←	Description du plan de la dernière recherche exécutée

La commande Lire dernier plan recherche retourne la description interne du plan d'exécution prévu pour la dernière recherche effectuée sur les données. Pour plus d'informations sur les descriptions de recherches, reportez-vous à la documentation de la commande [DECRIRE EXECUTION RECHERCHE](#).

Cette description est retournée au format Texte ou XML en fonction de la valeur passée dans le paramètre *formatDesc*. Vous pouvez passer une des constantes suivantes, placées dans le thème "Recherches" :

Constante	Type	Valeur
Description format Texte	Entier long	0
Description format XML	Entier long	1

Cette commande retourne une valeur significative si la commande **DECRIRE EXECUTION RECHERCHE** a été exécutée au cours de la session.

La description du plan de la dernière recherche peut être comparée à la description du chemin réel de la dernière recherche (obtenue à l'aide de la commande **Lire dernier chemin recherche**) à des fins d'optimisations. Pour plus d'informations, reportez-vous à la description de la commande **DECRIRE EXECUTION RECHERCHE**.

Lire dernier chemin recherche

Lire dernier chemin recherche (formatDesc) → Chaîne

Paramètres	Type	Description
formatDesc	Entier long →	Format de description : Texte ou XML
Résultat	Chaîne ←	Description du chemin de la dernière recherche exécutée

La commande **Lire dernier chemin recherche** retourne la description interne détaillée du chemin réel de la dernière recherche effectuée sur les données. Pour plus d'informations sur les descriptions de recherches, reportez-vous à la documentation de la commande **DECRIRE EXECUTION RECHERCHE**.

Cette description est retournée au format Texte ou XML en fonction de la valeur passée dans le paramètre *formatDesc*. Vous pouvez passer une des constantes suivantes, placées dans le thème "Recherches" :

Constante	Type	Valeur
Description format Texte	Entier long	0
Description format XML	Entier long	1

Cette commande retourne une valeur significative si la commande **DECRIRE EXECUTION RECHERCHE** a été exécutée au cours de la session.

La description du chemin de la dernière recherche peut être comparée à la description du plan prévu de la dernière recherche (obtenue à l'aide de la commande **Lire dernier plan recherche**) à des fins d'optimisations. Pour plus d'informations, reportez-vous à la description de la commande **DECRIRE EXECUTION RECHERCHE**.

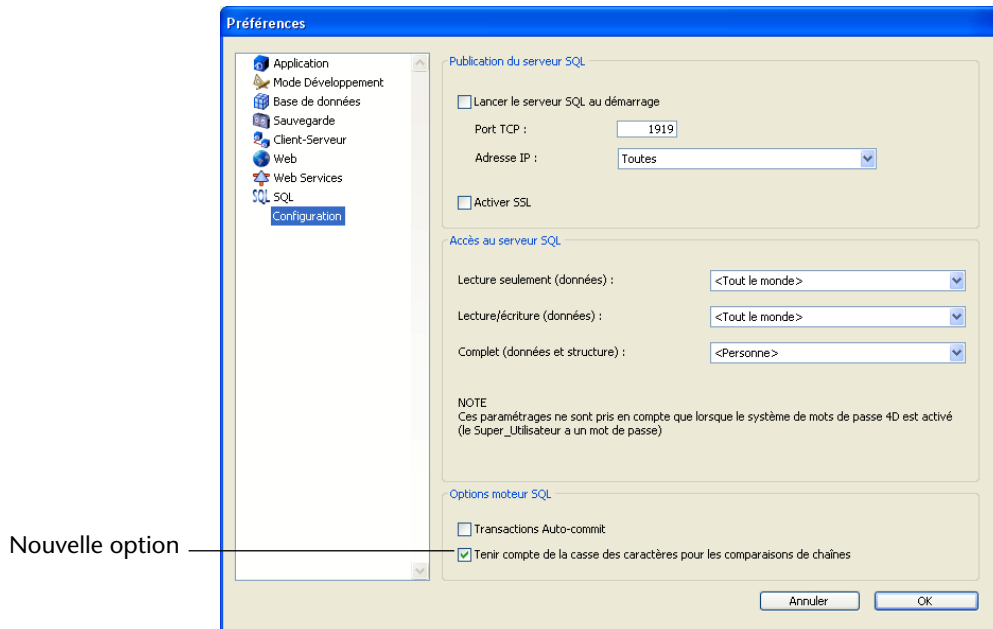
Moteur SQL

Les nouveautés décrites dans ce paragraphe concernent le moteur SQL intégré de 4D.

Gestion de la casse des caractères

Une nouvelle préférence permet de modifier le mode de prise en compte de la casse des caractères dans les requêtes SQL : “Tenir compte de la casse des caractères pour les comparaisons de chaînes”.

L’option est située dans la page SQL/Configuration des Préférences :



Cette option est cochée par défaut, ce qui signifie que le moteur SQL établit une différence entre les majuscules et les minuscules lors des comparaisons de chaînes (tris et recherches). Par exemple “ABC”= “ABC” mais “ABC” # “Abc”.

Dans certains cas, par exemple pour aligner le fonctionnement du moteur SQL sur celui du moteur 4D, vous pourrez souhaiter que les comparaisons de chaînes ne tiennent pas compte de la casse (“ABC”=“Abc”). Pour cela, il suffit de désélectionner l’option.

**Nouveau sélecteur pour
FIXER PARAMETRE BASE
et Lire parametre base**

L'option de prise en compte de la casse par le moteur SQL peut être fixée et lue par programmation à l'aide des commandes FIXER PARAMETRE BASE et Lire parametre base. Pour cela, un nouveau sélecteur a été ajouté :

- **Sélecteur = 44** (Casse caractères moteur SQL)
 - Valeurs possibles : 0 ou 1 (0 = casse non prise en compte, 1 = casse prise en compte)
 - Description : Activation ou inactivation de la prise en compte de la casse des caractères pour les comparaisons de chaînes effectuées par le moteur SQL.

**Référencer des
expressions de type
pointeur**

Il est désormais possible de référencer directement des expressions de type pointeur (dépointé ou non) dans les requêtes SQL.

- Passage d'une variable par un pointeur dépointé :

```
C_ENTIER LONG($vLong)
C_POINTEUR($vPointeur)
$vLong:=1
$vPointeur:=->$vLong
Debut SQL
  SELECT Col1 FROM TEST WHERE Col1=:$vPointeur->;
Fin SQL
```

- Passage d'une variable par un pointeur non dépointé :

```
C_ENTIER LONG($vLong)
C_POINTEUR($vPointeur)
$vLong:=1
$vPointeur:=->$vLong
Debut SQL
  SELECT Col1 FROM TEST WHERE Col1=:$vPointeur;
Fin SQL
```

Ce principe fonctionne avec tous les types de requêtes SQL : commandes ODBC, balises Debut/Fin SQL et commande CHERCHER PAR SQL.

Note Seul un niveau de pointeur est pris en charge ; les pointeurs de pointeurs ne peuvent pas être utilisés.

Unicode

4D v11 SQL Release 1 comporte plusieurs nouveautés liées à la prise en charge étendue de l'Unicode.

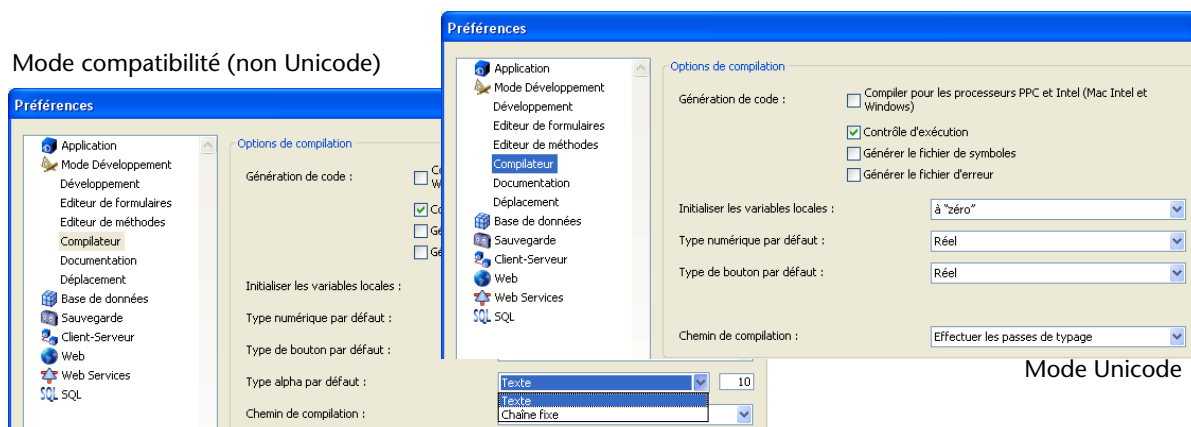
Menus

Dans une application 4D, les menus et leurs libellés peuvent désormais être affichés en Unicode. Cette possibilité permet d'utiliser par exemple des caractères japonais et grecs dans le même menu. Cette nouveauté concerne tous les types de menus, qu'ils aient été créés via l'éditeur de menus ou par programmation.

Bien entendu, cette fonction est disponible uniquement lorsque le mode Unicode est activé dans la base.

Option Type alpha par défaut

L'option de compilation "Type alpha par défaut" de la page Mode Développement/Compilateur est supprimée lorsque la base fonctionne en mode Unicode :



En mode Unicode, le type Texte est automatiquement utilisé.

Remplacer chaîne

Remplacer chaîne (source; obsolète; nouveau{; remplacements}{; *}) →
Chaîne

Paramètres	Type	Description
source	Chaîne	→ Chaîne de départ
obsolète	Chaîne	→ Caractère(s) à remplacer
nouveau	Chaîne	→ Chaîne de remplacement
remplacements	Numérique	→ Nombre de remplacements à effectuer
*	*	→ <i>Si passé : évaluation diacritique</i>
Résultat	Chaîne	← Chaîne résultante

La commande Remplacer chaîne admet désormais l'astérisque * en dernier paramètre. Si vous passez ce paramètre, vous indiquez que l'évaluation des caractères doit être diacritique, c'est-à-dire qu'elle doit tenir compte de la casse des caractères et des caractères accentués (a#A, a#à...).

Ce principe est illustré dans les exemples suivants :

```
vRésultat:=Remplacer chaîne("Crème brûlée";"Brulee";"caramel")
`vRésultat est égal à "Crème caramel"
```

```
vRésultat:=Remplacer chaîne("Crème brûlée";"Brulee";"caramel";*)
`vRésultat est égal à "Crème brûlée"
```

4D View

Dans la nouvelle version du plug-in 4D View, la saisie et l'affichage de caractères s'effectue désormais en Unicode. Ce mode est actif quel que soit l'état de l'option "Unicode" de 4D.

4D View v11.1 peut ouvrir les documents créés en version précédente. En revanche, les documents 4D View créés à partir de la version 11.1 ne pourront pas être rouverts en version 11 ou 2004.x.

Autres nouveautés

Etats rapides et paramètres d'impression

Pour maintenir la cohérence de vos interfaces, 4D uniformise désormais les paramètres d'impression des zones d'états rapides et celles de l'application.

Dans les versions précédentes de 4D, les paramètres d'impression des zones d'états rapides étaient stockés séparément avec chaque état. Les éventuelles modifications des paramètres d'impression de 4D n'étaient pas reportées sur les états rapides et inversement.

A compter de la version 11.1, les paramètres d'impression sont toujours identiques dans 4D et dans les zones d'états rapides. Toute modification effectuée sur les paramètres d'impression de 4D est reportée sur ceux des états rapides. A l'inverse, lors du chargement d'un état rapide, les paramètres d'impression de l'état remplacent ceux de 4D.

Coordonnées du clic sur une image

4D permet désormais de récupérer les coordonnées locales d'un clic dans un champ ou une variable image, même si un défilement ou un zoom a été appliqué à l'image.

Les coordonnées du clic sont retournées dans les variables système *MouseX* et *MouseY*. Les coordonnées sont exprimées en pixels par rapport à l'angle supérieur gauche de l'image (0,0). Vous devez lire la valeur de ces variables dans le cadre de l'événement formulaire Sur clic ou Sur double clic.

Pour que ce mécanisme fonctionne correctement, le format d'affichage doit être "Image tronquée (non centrée)".

Ce mécanisme, proche de celui d'une *image map*, peut être utilisé par exemple pour gérer des barres de boutons défilables ou l'interface de logiciels de cartographie.