



Forum: WD 7.x

Topic: SQLManagerX V4

Subject: SQLManagerX V4

Posté par: Anonyme

Contribution le : 27/9/2005 11:39:46

Bonjour a tous

Après 8 mois de travail nous avons enfin terminé la version 4
cette version a demandé beaucoup de travail car on a retravaillé sur tous les accès natifs et sur une
revue de code de SQLManagerX pour l'harmoniser avec tous les accès.

nouveautés :

Gestion des blobs pour MySQL, SQLite, FB
bientôt OTL, PostgreSQL et HF

SQLAttacheMemo

pour insérer un fichier image, txt, word, excel etc....

dans un membre ensuite la SQLManagerX l'insérera dans la base avec insert ou update

SQLChargeMemo

pour afficher une colonne memo dans un champ de la fenêtre (image txt, html, etc....)

SQLTableConstruit ()

SQLTableConstruit permet de créer une table mémoire dynamiquement en fonction des filtres et des
colonnes à renvoyer. elle

ATTENTION :

le paramètre XLPAR_listeCol est très important. En effet il détermine quelle méthode va être utilisée
pour charger la table si ce paramètre est vide c'est la méthode Ctable qui est utilisée et donc toutes
les lignes vont être chargées. Donc pour certaines requêtes cela risque d'être lent.

Il vaut mieux privilégier Xtable en donnant le nom des colonnes ou bien avoir un filtre permettant de
ne remonter que quelques lignes dans la table mémoire.

SQLTableEdit

Permet d'éditer une table préalablement chargée par SQLXtable ou SQLCtable.

avec une seule ligne de code

GetSchema ()

Permet d'avoir sous forme de chaîne, la création de la table et des index
dans 2 membres :schemaTable et :schemaIndex

GetPrimaryKey ()

Permet de connaître les colonnes faisant partie de la primary key de la table

SetPrimaryKey ()

Permet de modifier les colonnes faisant partie de la primary key de la table

Pour PERVASIVE cette fonction est obligatoire dans la déclaration de l'objet. La base ne renvoyant
pas les colonnes qui font partie de la primary key il faut alors au moment de la déclaration rajouter

une ligne SetPrimaryKey pour que SQLManagerX fonctionne correctement

GetAutoIncrement()

Permet de connaître les colonnes qui sont autoIncrement dans la table

SetAutoIncrement ()

SQLManagerX gère les autoIncrement en n'envoyant pas la valeur dans la base (cette dernière faisant l'increment et ajoutant dans la table la valeur) cette valeur est normalement stocke dans :dernierID

il n'y a pas vraiment de cas précis ou on a besoin de modifier les autoIncrement. Cette fonction est plutôt là au cas ou SQLManagerX n'arriverai pas pas déterminé que la colonne est auto Increment dans votre base.

tous cela disponible sur www.SQLManagerX.com
dans le menu Acces natif ou SQLManageX