

c_tableSL - Utilisation

Version 1.0.0.1

Rodolphe Jouannet

24 janvier 2008

Table des matières

0.1	Introduction	2
0.2	Description des membres	2
0.2.1	currentRec	2
0.2.2	requestNumber	2
0.2.3	queryString	3
0.2.4	genericOrder	3
0.2.5	genericWhere	3
0.2.6	tableName	3
0.3	Description des méthodes	3
0.3.1	Constructeur()	3
0.3.2	AfficheTable()	3
0.3.3	GereTouche()	3
0.3.4	TableInit()	4
0.3.5	TriTable()	4

0.1 Introduction

La classe `c_tableSQL` permet de gérer automatiquement, à la façon d'un R.A.D., la consultation d'enregistrements en mode table, provenant d'une clause **SELECT** quelconque. Cette classe permet de n'afficher que les "n" lignes apparentes de la table `Windev` et, à l'aide d'un ascenseur, de pouvoir afficher les "n" lignes suivantes ou précédentes, sans pour cela remonter en mémoire la totalité des enregistrements renvoyés par la clause **SELECT**.

0.2 Description des membres

0.2.1 currentRec

Ce membre contient le n° d'enregistrement en cours. Peut être initialisé à 0 (ou -1) pour afficher la table depuis le premier enregistrement.

0.2.2 requestNumber

Ce membre contient le numéro de requête en cours.

0.2.3 queryString

Ce membre contient la requête complète envoyer au serveur MySQL au travers de la classe `c_tableSQL`.

0.2.4 genericOrder

Ce membre contient l'ordre de tri initial, et est modifié par la méthode `TriTable()`.

0.2.5 genericWhere

Ce membre contient la clause **WHERE** initial et doit être modifiée pour affecter le contenu des enregistrements affichés dans la table Windev.

0.2.6 tableName

Ce membre contient le nom de la table Windev qui contiendra les enregistrements de la base à afficher.

0.3 Description des méthodes

0.3.1 Constructeur()

L'instanciation de la classe `c_tableSQL` nécessite obligatoirement 3 paramètres :

- le nom de l'instance de la classe `mySQL`,
- le nom de la table Windev recevant les enregistrements,
- le nom de l'ascenseur dans la fenêtre Windev, lié à la table précédemment définie,

Exemple d'instanciation :

```
maTable is c_tableSQL(mSQL, "tbl_maTable", "asc_maTable")
```

0.3.2 AfficheTable()

Cette méthode permet d'afficher les enregistrements de la clause **SELECT** dans la table Windev, en ne chargeant en mémoire que le nombre de ligne réellement affichée à l'écran. Deux paramètres sont nécessaire, `p_readUncommitted` qui permet de savoir si l'on doit démarrer une transaction ou non, et `p_currentRec` qui permet de se positionner à un n° d'enregistrement particulier.

Exemple d'appel de la méthode :

```
maTable :AfficheTable(True, -1)
```

0.3.3 GereTouche()

Cette méthode permet de gérer les actions à effectuer suite à l'appuie de la touche "flèche haut" ou "flèche bas", ou l'appuie sur le champs "scrollBar". Cette méthode doit être appelée depuis la table Windev, procédure "Touche enfoncée (WM_KEYDOWN)".

Exemple d'appel de la méthode :

```
maTable :GereTouche(_EVE.wParam)
```

0.3.4 TableInit()

Cette méthode est utilisée en interne par la classe `c_tableSQL` et ne doit pas être appelée de l'extérieur.

0.3.5 TriTable()

Cette méthode doit être appelée de boutons internes à la fenêtre contenant la table visualisant les enregistrements, et elle permet de modifier la clause **WHERE** de l'instance de la classe afin de trier certains champs en mode ASC ou DESC.

Exemple d'appel de la méthode :

```
maTable :TriTable(ChampsEnCours(), "client.client_ID")
```