



# Tables .csv dynamiques avec Base



*Version 1.0 du 18.06.2006*

Réalisé avec : **OOo 2.0**

Plate-forme / Os : **Toutes**

*Distribué par le projet fr.OpenOffice.org*

## Sommaire

1 Présentation.....	<a href="#">3</a>
2 Pré-requis.....	<a href="#">3</a>
3 Création d'une table Texte.....	<a href="#">3</a>
4 Liaison dynamique avec un fichier .csv.....	<a href="#">4</a>
5 Les options.....	<a href="#">4</a>
5.1 Options d'encodage.....	<a href="#">4</a>
5.2 Options de séparateur de champs.....	<a href="#">4</a>
5.3 Options avancées de séparateur de champs.....	<a href="#">5</a>
5.4 Option en-tête de colonnes.....	<a href="#">5</a>
5.5 Options de guillemets.....	<a href="#">5</a>
5.6 Option DESC.....	<a href="#">5</a>
6 Annexe : importation d'une table issue d'un fichier .csv.....	<a href="#">6</a>
7 Crédits .....	<a href="#">7</a>
8 Licence.....	<a href="#">7</a>

## 1 Présentation

Le but de ce document est de présenter une technique permettant d'utiliser de manière dynamique des fichiers de données au format .csv dans un document Base, éventuellement avec des tables intégrées au document. Cela permet de :

- ◆ Utiliser dans Base des données issues d'une autre application qui génère des fichiers .csv.
- ◆ Partager des données saisies sous Base avec une autre application capable de traiter des données au format .csv.

Dans la suite de ce document, une table dont les données sont stockées dans un fichier texte sera désignée comme table Texte. Toutes les instructions SQL indiquées doivent être exécutées par l'outil de saisie de commandes SQL directes, sauf indication contraire.

**Attention** : Cette technique fonctionne uniquement avec une base intégrée HSQLDB.

## 2 Pré-requis

Ce document fait usage de commandes SQL exécutées grâce à l'outil de saisie de commandes SQL directes (Menu Outils > SQL ...) proposé par Base. Une première présentation de cet outil peut être trouvée dans le document FAQ SQL Direct ( [http://fr.openoffice.org/Documentation/How-to/Bdd/FAQ\\_SQL\\_direct.odt](http://fr.openoffice.org/Documentation/How-to/Bdd/FAQ_SQL_direct.odt)). Une connaissance de base des instructions SQL sera utile pour profiter de ce document, les commandes permettant la création d'une table ne seront pas développées ici.

La référence des commandes SQL utilisées est la documentation de HSQLDB disponible à l'adresse <http://hsqldb.org/web/hsqldbDocsFrame.html>, et en particulier la partie qui concerne les tables Texte.

## 3 Création d'une table Texte

Pour créer une table Texte dans un document Base, il faut exécuter une instruction spécifique :

```
CREATE TEXT TABLE <nomTable> (<définitionDeColonne>
[<définitionDeContrainte>])
```

Pour connaître la syntaxe et les instructions de <définitionDeColonne> et de <définitionDeContrainte>, référez vous à la documentation HSQLDB.

**Attention** : Sous Base, il faut effectuer une actualisation des tables par le menu Afficher > Actualiser les tables pour voir apparaître les tables ainsi créées.

Si la table doit être liée à un fichier .csv déjà existant, il faut évidemment que les types de données des champs de la table correspondent aux types de données stockées dans le fichier .csv.

Soit un fichier .csv se présentant ainsi :

```
1,durand,jean-marc
2,leroy,pierre
3,martin,nicole
```

Une instruction possible pour créer une table cohérente avec ce fichier est :

```
CREATE TEXT TABLE "TableCSV" ("id" INTEGER, "nom" VARCHAR(50), "prenom"
VARCHAR(50))
```

Si vous souhaitez pouvoir saisir des données dans cette table Texte (et donc modifier le fichier .csv) depuis Base, la table doit comprendre une clé primaire. Par exemple :

```
CREATE TEXT TABLE "TableCSV" ("id" INTEGER PRIMARY KEY, "nom" VARCHAR(50),
"prenom" VARCHAR(50))
```

## 4 Liaison dynamique avec un fichier .csv

Pour établir la liaison dynamique avec un fichier .csv, il faut utiliser l'instruction suivante :

```
SET TABLE <nomTable> SOURCE <cheminRelatifDuFichier_etOptions>
```

Il n'est pas nécessaire que le fichier lié soit nommé avec une extension .csv. Un fichier nommé truc.txt est parfaitement valide.

**Attention** : Le fichier lié doit impérativement être soit dans le même répertoire que le fichier Base, soit dans un sous-répertoire du répertoire du fichier Base. Si le fichier lié n'existe pas encore, il faut créer un fichier vide portant le nom souhaité avant d'exécuter cette instruction. D'autre part, tant que le document Base est ouvert, le fichier lié n'est (normalement) accessible qu'en lecture seule par les autres applications. Si le fichier lié est modifié pendant que le document Base est ouvert, les modifications n'apparaissent pas dans le document Base. Il faut alors le fermer et le rouvrir pour voir les modifications dans Base.

Si le fichier lié est dans le même répertoire que le fichier Base, l'instruction se présente ainsi (sans options) :

```
SET TABLE "MaTable" SOURCE "monFichier.csv"
```

Si le fichier lié est dans le sous-répertoire /Truc du répertoire contenant le fichier Base, l'instruction se présente ainsi :

```
SET TABLE "MaTable" SOURCE "/Truc/monFichier.csv"
```

## 5 Les options

### 5.1 Options d'encodage

L'option d'encodage (encoding= ) permet de spécifier quel encodage de caractères est utilisé par le fichier lié. Sa valeur par défaut est ASCII. Les autres encodages disponibles peuvent varier suivant l'environnement Java, mais les encodages suivants doivent être présents quel que soit l'environnement Java :

- ◆ US-ASCII                    ASCII 7-bit, alias ISO646-US, alias bloc Latin Basic du jeu de caractères Unicode.
- ◆ ISO-8859-1                ISO Alphabet Latin No. 1, alias ISO-LATIN-1
- ◆ UTF-8                      Unicode 8-bit
- ◆ UTF-16BE                 Unicode 16-bit, big-endian byte order
- ◆ UTF-16LE                 Unicode 16-bit, little-endian byte order
- ◆ UTF-16                    Unicode 16-bit, byte order spécifié par une marque de byte order initiale obligatoire. (Les deux byte orders sont acceptés en lecture, big-endian seulement à l'écriture)

Par exemple, pour spécifier un encodage UTF-8, utiliser l'instruction suivante :

```
SET TABLE "MaTable" SOURCE "monFichier.csv;encoding=UTF-8"
```

### 5.2 Options de séparateur de champs

Les options de séparateur de champs (fs = ) permettent de spécifier quel caractère est utilisé pour séparer les champs dans le fichier lié. Sa valeur par défaut est ',' (virgule). Pour spécifier un autre caractère, utiliser l'option fs. Par exemple, pour un fichier lié dont le séparateur est le point '.', utiliser l'instruction suivante :

```
SET TABLE "MaTable" SOURCE "monFichier.csv;fs=."
```

Certains caractères séparateurs disposent d'un indicateur spécial :

- ◆ \semi            double point ':'

- ◆ \space    espace ' '
- ◆ \t        tabulation
- ◆ \\        backslash '\'
- ◆ \u####   caractère Unicode exprimé en hexadécimal

Par exemple, si le séparateur de champs est la tabulation, utiliser l'instruction suivante :

```
SET TABLE "MaTable" SOURCE "monFichier.csv;fs=\t"
```

### 5.3 Options avancées de séparateur de champs

Il est possible de spécifier des séparateurs de champs différents pour les champs de type VARCHAR (vs= ) et LONGVARCHAR (lvs= ).

Par exemple, pour une table dont le premier champ est de type CHAR, le deuxième de type VARCHAR, et pour laquelle on a attribué la valeur '|' à fs et la valeur '.' à vs, une ligne du fichier lié se présentera ainsi :

```
donneesChamp1|donnéesChamp2.autresDonnées
```

Pour attribuer ces valeurs, utilisez une instruction de cette forme :

```
SET TABLE "MaTable" SOURCE "monFichier.csv;fs=|;vs=."
```

### 5.4 Option en-tête de colonnes

Si la première ligne du fichier lié contient des en-têtes de colonnes, l'option ignore\_first=true permet d'indiquer qu'il faut ignorer la première ligne. Par exemple :

```
SET TABLE "MaTable" SOURCE "monFichier.csv;ignore_first=true"
```

La valeur par défaut de ignore\_first est false.

### 5.5 Options de guillemets

Il existe deux options concernant l'encadrement des chaînes de caractères par des guillemets :

- ◆ all\_quoted (valeur par défaut = false) : indique que tous les champs contenant une chaîne de caractères doivent être entourés de guillemets simples lors de l'écriture dans le fichier lié.
- ◆ quoted (valeur par défaut = true) : indique que les champs contenant une chaîne de caractères seront entourés de guillemets simples lors de l'écriture dans le fichier lié seulement si ils contiennent le caractère séparateur de champs.

Lorsque ces deux options sont à leur valeur par défaut (all\_quoted = false, quoted = true), les champs contenant des chaînes de caractères seront écrits dans le fichier lié avec des guillemets simples uniquement si ils contiennent des guillemets ou le caractère séparateur de champs. Le guillemet simple est doublé lorsqu'il est contenu dans la chaîne de caractères.

Lorsque all\_quoted = false et quoted = false, il n'est pas possible d'insérer un champ contenant une chaîne de caractères si la chaîne contient le caractère séparateur de champs car il serait alors impossible de distinguer ce caractère du séparateur de champs.

Lors de la lecture du fichier lié, le programme traite chaque champ individuellement. Un champ est considéré comme encadré de guillemets si le premier caractère du champ est un guillemet, et il est alors traité selon ce critère.

### 5.6 Option DESC

L'option DESC permet de spécifier que le fichier lié est en lecture seule pour Base et garantit également que le fichier lié sera intégralement relu lors de chaque commande SELECT. L'option DESC s'utilise ainsi :

```
SET TABLE "MaTable" SOURCE "monFichier.csv" DESC
```

Lorsque cette option est utilisée, il n'est pas possible de changer ultérieurement le statut « lecture

seule » de la table.

## 6 Annexe : importation d'une table issue d'un fichier .csv

L'utilisation des tables Texte permet l'importation de données issues d'un fichier .csv. Il s'agit ici d'effectuer une importation non dynamique sans utiliser le copier/coller proposé par OOO 2 qui semble relativement instable.

Le principe consiste en la création d'une table Texte temporaire dont les données sont ensuite insérées dans une table intégrée classique.

Voici un exemple d'instructions (à exécuter successivement) dans le cas d'une table comportant un champ id (INTEGER PRIMARY KEY) et un champ nom (VARCHAR(50)) :

```
CREATE TEMP TEXT TABLE "tableTemporaire" ("id" INTEGER, "nom" VARCHAR(50))
SET TABLE "tableTemporaire" SOURCE "leFichierSource.csv"
CREATE TABLE "tableFinale" ("id" INTEGER PRIMARY KEY, "nom" varchar(50))
INSERT INTO "tableFinale" ("id", "nom") SELECT "id", "nom" FROM
"tableTemporaire"
commit
```

## 7 Crédits

Auteur : **Manuel NAUDIN**

Remerciement : **Tony GALMICHE et tous les relecteurs de la liste documentation.**

Intégré par : **Tony GALMICHE**

Contacts : **Projet Documentation OpenOffice.org - [fr.OpenOffice.org](http://fr.OpenOffice.org)**

Traduction :

Historique des modifications :

<b>Version</b>	<b>Date</b>	<b>Commentaire</b>
1.0	18/06/06	Création

## 8 Licence

### Appendix

#### Public Documentation License Notice

The contents of this Documentation are subject to the Public Documentation License Version 1.0 (the "License"); you may only use this Documentation if you comply with the terms of this License. A copy of the License is available at <http://www.openoffice.org/licenses/PDL.html>.

The Original Documentation is Tables .csv dynamiques avec Base. The Initial Writer of the Original Documentation is Manuel Naudin Copyright © 2006. All Rights Reserved. (Initial Writer contact(s): audionuma@gmail.com).

Contributor(s): \_\_\_\_\_.  
Portions created by \_\_\_\_\_ are Copyright © \_\_\_\_\_ *[Insert year(s)]*. All Rights Reserved.  
(Contributor contact(s): \_\_\_\_\_ *[Insert hyperlink/alias]*).

NOTE: The text of this Appendix may differ slightly from the text of the notices in the files of the Original Documentation. You should use the text of this Appendix rather than the text found in the Original Documentation for Your Modifications.