



Extrait du Environnement iSeries

<http://xdocs400.com/spip.php?article410>

# Résultat d'une requete SQL au format CSV ou XML dans un fichier de l'IFS

- Les articles -



Date de mise en ligne : mardi 31 janvier 2012

Date de parution : 14 décembre 2007

## **Description :**

Pour ne pas avoir à recoder vos exports de données à chaque nouvelle requête sql, utilisez la SQLDA.

---

**Environnement iSeries**

---

**Cet article vous propose d'utiliser la SQLDA pour bénéficier de la puissance d'un vrai système de requêtes SQL dynamiques.**

**Passez n'importe quelle requête à ce programme pour obtenir une sortie au format CSV ou XML dans l'IFS.**

**Cet article est surtout prétexte à présenter le fonctionnement globale de l'utilisation de SQL avec la SQLDA : jetez donc un coup d'oeil aux programmes DB2TOCSV et DB2TOXML.**

**Dans un premier temps cet article présentait DB2TOCSV et DB2TOXML qui font appel à des programmes de services. Je laisse les exemples en ligne, car l'utilisation de programmes de services est un très bon choix car les composants applicatifs sont partageables.**

**Mais en pensant à tous ceux qui souhaitent emporter un outil facilement chez un client, j'ai intégré DB2TOCSV, DB2TOXML et les programmes de service dans un seul programme RPG : DB2TOIFS (avec une commande de lancement et un programme de vérification de la commande).**

**Malgré une longue expérience, je suis un débutant en RPG. Soyez indulgent vis à vis de mon codage. Toutes les idées sont les bienvenues.**

**Et merci à tous ceux qui reconnaîtrons sûrement des bouts de leurs propres codes.**

## Instructions

- Les scripts d'installation sont proposés en document joint au format ZIP. Extraire le fichier suffixé par XML et le sauvegarder dans un répertoire de votre IFS.
- Utilisez l'outil d'installation XML2SRC (voir [Outil d'installation des programmes mis en ligne sur XDOCS400](#), pour installer l'outil ou consulter le mode opératoire d'une installation).
- Pour installer DB2TOCSV et DB2TOXML, il faut au préalable avoir installé les programmes de services que vous trouverez sur l'article [Construire des programmes de services utiles au quotidien](#)
- A compter du 12/03/2009, Seul DB2TOIFS est maintenu.

## Limitations

- Votre AS400 doit être en V5R4.

## Notez bien

- Le site XDOCS400.COM fournit des programmes à titre d'exemples et n'assure pas de hotline ou de garantie dessus.
- Toutefois, si vous avez des soucis pour les installer, ou les utiliser, contactez l'auteur de l'article via le formulaire ad'hoc du site xdocs400 pour une aide ponctuelle.

## Fonctionnement de l'export au format CSV ou XML via DB2TOIFS

- DB2TOIFS est une version sans programmes de services. Tout est inclus dans le source, pour faciliter la vie de ceux qui se déplacent en clientèle et qui veulent emmener un seul objet.
- Un paramètre supplémentaire est ajouté : le format d'export \*XML ou \*CSV.
- Le CCSID 1252 est indiqué par défaut
- Au format XML, seuls les paramètres requête, nom du fichier ifs, ccsid et séparateur décimal sont pris en compte. Les autres ne sont utilisés que pour le format csv.
- Le fichier XML généré contient expressément iso-8859-1 qui correspond à l'encodage de l'Europe de l'Ouest.

### Exemple d'utilisation : exécution en ligne de commande pour un export XML

```
DB2TOIFS REQUETE('select * from table')
DOCUMENT('/home/repertoire/extraction.xml')
FORMAT(*XML)
```

### Exemple d'utilisation : exécution en ligne de commande pour un export CSV

```
DB2TOIFS REQUETE('select * from table')
DOCUMENT('/home/repertoire/extraction.csv')
FORMAT(*CSV)
```

---

```
ENTETE(O)
```

## Fonctionnement de l'export au format CSV (avec programmes de services)

- DB2TOCSV utilise des programmes de services et des clauses COPY qui doivent être installés. Ils sont décrits dans cet article : [Construire des programmes de services utiles au quotidien](#)

### L'utilitaire s'utilise via la commande DB2TOCSV qui utilise les paramètres :

- Requête SQL
- Document au format CSV dans l'IFS
- Mode de création (\*REPLACE ou \*ADD)
- Caractère de fin de ligne (CRLF par défaut)
- Caractère séparateur de champ (le ; par défaut)
- Le caractère séparateur de décimales (, par défaut)
- Chaque champ est automatiquement encadré par des quote (N par défaut)
- Inclure une 1ère ligne qui contient les noms des colonnes ramenée par la requête SQL (O par défaut)
- CCSID du fichier dans l'IFS

### Exemple d'utilisation : exécution en ligne de commande

```
DB2TOCSV REQUETE('select * from table')
DOCUMENT('/home/repertoire/extraction.csv')
ENTETE(0)
CCSID(1208)
```

## Fonctionnement de l'export au format XML (avec programmes de services)

- DB2TOXML utilise des programmes de services et des clauses COPY qui doivent être installés. Ils sont décrits dans cet article : [Construire des programmes de services utiles au quotidien](#)

Il s'agit d'un export au format xml vraiment simplifié, facilement exploitable par excel. Exemple de structure après lancement de l'outil :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
ART
Fichiers Phy (F22)
PPDxxxD
55
```

```
ART
Fichiers Phy (F22)
PPDxxx0
136
```

**L'utilitaire s'utilise via la commande DB2TOXML qui utilise les paramètres :**

- Requête SQL
- Document au format XML dans l'IFS
- Le caractère séparateur de décimales (, par défaut)
- CCSID du fichier dans l'IFS

Le fichier est systématiquement recréé

**Exemple d'utilisation : exécution dans un programme avec paramétrage**

```
DB2TOXML REQUETE(&RES) DOCUMENT(&DOC) DECSEP(&SEP) +
CCSID(&CCSID)
```

---

## Description très générale des programmes DB2TOCSV et DB2TOXML et donc de DB2TOIFS qui fait les deux

Ces deux programmes ont un fonctionnement équivalent, hormis leur destination CSV ou XML.

## Les API C pour gérer l'IFS

- Les API C de l'IFS sont utilisées pour créer, ouvrir, écrire fermer les document de l'IFS.
- Création du document dans l'IFS avec le CCSID voulu (En XML, le document est systématiquement supprimé puis recréé.)
- L'ouverture d'un document de l'IFS en mode texte force le système à une conversion implicite.

## Structure de la SQLDA

- SQLDAID, Chaîne "SQLDA"
- SQLDABC, Longueur de la structure :  $SQLN * 44 + 16$
- SQLN, Nombre total d'occurrences SQLVAR allouées, égal au nombre de colonnes ramenées par la requête
- SQLD, Nombre d'occurrences SQLVAR utilisées.
- SQLVAR, Structure de données dont le nombre d'occurrences est SQLD
- SQLTYPE, Nombre qui représente le type de données de la colonne ou de la variable hôte. Un nombre impair indique qu'une valeur null est autorisée.
- SQLLEN, Longueur d'une valeur de colonne. Pour une valeur décimale, le premier octet représente la longueur de la partie entière et le second, la précision.
- SQLDATA, Utilisée pour les FETCH, OPEN et EXECUTE. L'application doit y placer l'adresse de la variable hôte qui reçoit les valeurs. Non utilisée par les instructions DESCRIBE and PREPARE.
- SQLIND, Utilisée pour les FETCH, OPEN et EXECUTE. Adresse de l'indicateur associé à la variable s'il en existe un. Si une valeur null est autorisée, SQLIND est à -1 si la valeur est null. Elle est à 0 si la valeur n'est pas null. Non utilisée par les instructions DESCRIBE and PREPARE.
- SQLNAMELEN, Longueur du nom de la colonne
- SQLNAME, Nom de la colonne

## Roadmap d'utilisation d'un curseur de manière totalement dynamique via la SQLDA

- Déclaration du curseur
- Préparation de la requête
- Initialisation de la taille de la SQLDA pour n zones et du nombre de colonnes à 0
- Exécution de l'instruction DESCRIBE pour déterminer le nombre de colonnes ramenées par la requête (nSQLDA), la taille exacte de la SQLDA (szSQLDA) et allouer la mémoire nécessaire (pSQLDA).
- Calcul de la taille totale nécessaire pour le buffer (BufferSize) = somme de la taille allouée à chaque colonne ramenée par la requête
- Allocation de la mémoire nécessaire pour ramener un enregistrement complet (pMyData) en fonction de la taille totale calculée (BufferSize).
- Attribuer à chaque poste la position mémoire de la variable hôte qui accueille les données de chacune des colonnes ramenées par la requête.
- Ouverture du curseur
- Lecture de tous les enregistrements ramenés par la requête. Pour chacune des colonnes à traiter :
  - calcul de la longueur occupée dans le buffer, en fonction du type de la zone.
  - en fonction du type de la zone, appel de la fonction appropriée qui permet de restituer le contenu de la colonne de manière explicite et dans une chaîne de caractères.

## Petits trucs

- les lignes suivantes permettent de récupérer le label d'une colonne (ou entete de colonne)

```
pSQL_VAR = %addr(SQLVAR(i_Column+SQLD));
```

```
s_Label = SQL_NAME;
```

- Pour une colonne retournée par la requête, si le type de variable SQL (SQLTYPE) est impaire, cela signifie que les valeurs "null" sont autorisées dans la colonne. Dans ce cas on ajoute +2 à la position du pointeur (SQL\_IND & SQL\_DATA) car c'est l'espace réservé pour gérer l'indicateur Null.
- Les 2 premiers caractères d'une zone du type varchar sont réservés pour stockage de la longueur réelle du contenu. Il faut le prendre en compte lorsque l'on récupère les données dans la (SQL\_DATA). La valeur réelle est extraite à partir de la position du pointeur + 2 (SQL\_DATA+2) et pour la longueur du buffer pour la colonne - 2.

## Corrections

- 12/03/2009 :
  - Programme DB2TOIFS correction de la méthode de conversion pour fonctionnement correcte vers ccsid 1208 et tout autre ccsid pouvant être convertit à partir de l'EBCDIC (ccsid 297)
  - Programme DB2TOIFS lors de l'écriture du CRLF, seul le CR subsistait dans le fichier. Correction pour conserver CRLF.
  - Seul le programme DB2TOIFS sera maintenu à partir de cette version. DB2TOCSV et DB2TOXML n'évolueront plus.
- 07/07/2008 :
  - Programmes DB2TOIFS, DB2TOCSV et DB2TOXML pour intégrer les types de données 384, 385, 388, 389, 392, 393 (date, time et timestamp)
- 04/06/2008 : Evolution et bugss
  - Modification de DB2TOIFS, DB2TOCSV et DB2TOXML pour intégrer la gestion du type de données 453
  - Modification de DB2TOIFS pour gérer les caractères spéciaux en mode XML

## Pour compiler à partir de V6R1

Pour compiler à partir de la V6R1, cf [ce post du forum](#), modifier dans le source toutes les lignes :

```
Rc = write(Fd:%addr(s_Val)+2:%len(%trimr(s_Val)));
```

Par

```
Rc = write(Fd:%addr(s_Val:*data):%len(%trimr(s_Val)));
```

*Post-scriptum :*

*V5R4 minimum, car la partie SQL est codée en free rpg.*

## Résultat d'une requete SQL au format CSV ou XML dans un fichier de l'IFS

---

*La version actuelle des programmes DB2TOCSV et DB2TOXML ne peut écrire dans un fichier IFS de CCSID 1208 mais le programme DB2TOIFS a été modifié pour pouvoir le faire.*