



Extrait du Environnement iSeries

<http://xdocs400.com/spip.php?article396>

obtenir une liste d'objets AS400 via SQL

- Les articles -



Date de mise en ligne : jeudi 6 septembre 2007

Description :

Comment obtenir une liste d'objets AS400 via une requête SQL ? C'est possible via la création d'une UDTF

Environnement iSeries

Comment obtenir une liste d'objets AS400 via une requête SQL ? C'est possible via la création d'une UDTF.

Que fait la fonction UDTFOBJLST

La fonction UDTFOBJLST retourne une liste d'objets as400.

Fonctionnement

La fonction utilise un programme RPGLE qui liste les objets AS400 dans un USERSPACE via l'API QUSLOBJ.

Paramètres en entrée de la fonction

- Nom de l'objet (*ALL, NOM*, Nom de l'objet).
- Type de l'objet (*FILE, *PGM, ..., *ALL)
- Bibliothèque (Nom de la bibliothèque, *LIBL, *CURLIB, ...)

Paramètres en sortie de la fonction

- Bibliothèque de l'objet
- Nom de l'objet
- Type de l'objet
- Attribut étendu de l'objet
- Texte de l'objet
- Date + heure de création de l'objet
- Date + heure de modification de l'objet
- Utilisateur ayant créé l'objet

Création du programme de service

Télécharger le source du programme UDTFOBJLST et compilez le comme ci-dessous.

```
1) CRTRPGMOD MODULE(BIB/UDTFOBJLST)
   SRCFILE(BIB/QRPGLESRC)
```

```
2) CRTSRVPGM SRVPGM(BIB/UDTFOBJLST)
```

```
EXPORT(*ALL)
```

Création de la fonction SQL

```
CREATE FUNCTION UDTFOBJLST(VARCHAR(10), VARCHAR(10),VARCHAR(10))
RETURNS TABLE(
OBJ_LIB      CHAR(10),
OBJ_NAME     CHAR(10),
OBJ_TYPE     CHAR(10),
OBJ_EXTA     CHAR(10),
OBJ_TEXT     CHAR(50),
OBJ_CRTDTE   CHAR(19),
OBJ_CHGDTE   CHAR(19),
OBJ_CRTUSR   CHAR(10)
)
EXTERNAL NAME 'BIB/UDTFOBJLST(UDTFOBJLST)'
LANGUAGE RPGLE
DISALLOW PARALLEL
NO SQL
RETURNS NULL ON NULL INPUT
PARAMETER STYLE DB2SQL
NOT DETERMINISTIC
```

Exemple simple d'utilisation

```
select * from table(udtfobjlst('*ALL' , '*ALL' , '*CURLIB')) as liste
```

Exemples plus compliqués d'utilisation

A quoi peut bien servir une telle fonction, alors que l'on peut obtenir une liste d'objet via la commande DSPOBJD par exemple) ?

Essayez les requêtes ci-dessous, pour vous rendre compte de l'intérêt de ce type de fonction.

La requête ci-dessous liste tous les objets dont le nom + type se trouvent dans les deux bibliothèques BIB1 et BIB2.

```
SELECT * FROM
table(udtfobjlst('*ALL' , '*ALL' , 'BIB1')) as t1
inner join (
SELECT * FROM
table(udtfobjlst('*ALL' , '*ALL' , 'BIB2')) as liste
) as t2
on t2.obj_name = t1.obj_name
and t2.obj_type = t1.obj_type
```

La requête ci-dessous liste tous les objets dont le nom + type se trouvent dans la bibliothèque BIB1 mais pas dans BIB2.

```
SELECT * FROM
table(udtfobjlst('*ALL' , '*ALL' , 'BIB1')) as t1
exception join (
SELECT * FROM
table(udtfobjlst('*ALL' , '*ALL' , 'BIB2')) as liste
) as t2
on t2.obj_name = t1.obj_name
and t2.obj_type = t1.obj_type
```

La requête ci-dessous liste les objets de type fichier, en filtrant sur les fichiers physiques, l'utilisateur DAVID et en triant sur le nom de l'objet, puis la bibliothèque.

```
select * from table(udtfobjlst('*ALL' , '*FILE' , '*USRLIBL')) as liste
where obj_exta = 'PF'
and obj_user = 'DAVID'
order by obj_name, obj_lib
```

Avertissement

Le programme de service utilise l'API QUSLOBJ. Si vous passez en paramètres *ALL pour la liste de bibliothèques par exemple, le temps d'exécution devient très long et vous risquez fort d'obtenir un timeout. Votre requête n'aboutira pas. Dans ce cas, augmentez le paramètre UDF_TIME_OUT comme indiqué dans l'article [Créer une UDTF](#).

Post-scriptum :

Voir également l'article [Créer une UDTF](#).