



Extrait du Environnement iSeries

<https://xdocs400.com/spip.php?article458>

Contrôler l'existence d'un objet IBMi via SQL

- Les articles -



Date de mise en ligne : lundi 1er septembre 2014

Description :

Il peut être utile de disposer, en SQL, d'une fonction permettant de vérifier la présence d'un objet IBMi dans une bibliothèque, et ce quel que soit le type d'objet considéré.

Environnement iSeries

Contrôler l'existence d'un objet IBMi via SQL

Il peut être utile de disposer, en SQL, d'une fonction permettant de vérifier la présence d'un objet IBMi dans une bibliothèque, et ce quel que soit le type d'objet considéré.

Pour disposer de ce type de fonction en SQL, il nous faut créer un programme CL et 2 fonctions SQL (une première de type "externe", et la seconde de type "SQL") :

```
/* Source du CLP MABIBPGM/DWHSRC MBR(CHKOBJCLP) : */
/* Paramètres en entrée : */
/* Nom objet          10 */
/* Nom bibliothèque  10 */
/* Type d'objet       10 */
/* Paramètres en sortie : */
/* Trouve (1=oui ; 0=non) 1 */

PGM      PARM(&OBJET &BIBL &TYPE &TROUVE)
DCL      VAR(&OBJET) TYPE(*CHAR) LEN(10)
DCL      VAR(&BIBL) TYPE(*CHAR) LEN(10)
DCL      VAR(&TYPE) TYPE(*CHAR) LEN(10)
DCL      VAR(&TROUVE) TYPE(*CHAR) LEN(1)

/* Positionnement à "trouvé" par défaut */
CHGVAR   VAR(&TROUVE) VALUE('1')

/* Vérification présence programme */
CHKOBJ   OBJ(&BIBL/&OBJET) OBJTYPE(&TYPE)
MONMSG   MSGID(CPF9801) EXEC(DO)
CHGVAR   VAR(&TROUVE) VALUE('0')
ENDDO
ENDPGM
```

Voici le source de la fonction externe encapsulant le CL CHKOBJCLP pour vérification de l'existence d'un objet (dont le type IBMi est transmis en 3ème paramètre)

```
CREATE OR REPLACE FUNCTION MABIBPGM/CHKOBSQF (
  OBJET CHAR(10) ,
  BIBL CHAR(10) ,
  TYPOBJ CHAR(10)
)
RETURNS CHAR(1)
LANGUAGE CL
SPECIFIC MABIBPGM/CHKOBSQF
NOT DETERMINISTIC
MODIFIES SQL DATA
CALLED ON NULL INPUT
EXTERNAL NAME 'MABIBPGM/CHKOBJCLP'
PARAMETER STYLE SQL ;

LABEL ON SPECIFIC FUNCTION MABIBPGM/CHKOBSQF
IS 'Contrôle existence d'un objet IBMi' ;
```

Contrôler l'existence d'un objet IBMi via SQL

Voici maintenant le source de la fonction SQL destinée à faciliter l'appel de la fonction CHKOBJSQF, par l'utilisation de paramètre en entrée de type VARCHAR, alors que la fonction CHKOBJSQF ne tolère que le type CHAR, du fait qu'il s'agit d'une fonction externe encapsulant un CLP

```
CREATE OR REPLACE FUNCTION MABIBPGM/CHKOBJSQF (
  OBJET VARCHAR(10) ,
  BIBL VARCHAR(10) ,
  TYPOBJ VARCHAR(10)
)
RETURNS CHAR(1)
LANGUAGE SQL
SPECIFIC MABIBPGM/CHKOBJSQF
NOT DETERMINISTIC
MODIFIES SQL DATA
CALLED ON NULL INPUT
SET OPTION  ALWBLK = *ALLREAD ,
ALWCPYDTA = *OPTIMIZE ,
COMMIT = *NONE ,
DECRESULT = (31, 31, 00) ,
DFTRDBCOL = *NONE ,
DYNDFTCOL = *NO ,
DYNUSRPRF = *USER ,
SRTSEQ = *HEX
BEGIN
  DECLARE TOBJET CHAR ( 10 ) ;
  DECLARE TBIBL CHAR ( 10 ) ;
  DECLARE TTYPOBJ CHAR ( 10 ) ;
  SET TOBJET = OBJET ;
  SET TBIBL = BIBL ;
  SET TTYPOBJ = TYPOBJ ;

  RETURN
  WITH TMP AS (
    SELECT CHKOBJSQF ( TOBJET , TBIBL , TTYPOBJ ) AS TROUVE FROM SYSIBM / SYSDUMMY1
  )
  SELECT TROUVE FROM TMP ;
END ;

LABEL ON SPECIFIC FUNCTION MABIBPGM/CHKOBJSQF
IS 'Contrôle existence d'un objet IBMi' ;
```

A noter : vous trouverez un exemple d'utilisation de cet outil dans le dossier suivant, dans lequel j'explique comment [contrôler la validité des Jobd d'un système](#).

Post-scriptum :

Pour des digressions autour du développement en général, et du développement web en particulier, je vous invite à me retrouver sur mon blog <http://gregphplab.com>