



Extrait du Environnement iSeries

<http://xdocs400.com/spip.php?article259>

# Appel d'une procédure REXX à partir d'un programme RPG

- Les articles -



Date de mise en ligne : vendredi 25 novembre 2005

## **Description :**

Pour évaluer une expression numérique, il est très utile d'utiliser une procédure REXX. Mais comment l'appeler à partir d'un programme écrit en RPG, et récupérer la valeur calculée par la procédure REXX.

---

**Environnement iSeries**

---

Pour évaluer une expression numérique, il est très utile d'utiliser une procédure REXX. Mais comment l'appeler à partir d'un programme écrit en RPG, et récupérer la valeur calculée par la procédure REXX.

## Avertissements

Cet article contient des exemples de codes et non un utilitaire finalisé. Consultez la documentation IBM sur le REXX pour aller plus loin.

## Création de la procédure REXX

- Créer un membre CALCEXP dans le fichier source BIBSRC/QREXSRC et taper le code suivant dans le membre CALCEXP :

```
numeric digits 31
arg extarg
myval = "value =" extarg
interpret myval
push value
return 0
```

## Contenu du programme RPG

- Description de la procédure d'appel d'une commande CL :

```
*
* Pour exécution de commande CLP
DCommande      S          250
DCLPCMD        PR          EXTPGM('QCMDEXC')
D cmd          250
D cmdlen       15P 5 CONST
```

- Appel de la commande CL pour exécuter la procédure REXX :

```
/free
// -----
// Création du fichier de travail dans QTEMP.
// Commitment désactivé.
// -----
Commande = 'STREXPRC SRCMBR(CALCEXP) ' +
'SRCFILE(BIBSRC/QREXSRC) ' +
```

## Appel d'une procédure REXX à partir d'un programme RPG

```
'PARM(''125 + 69 * (12 / 3)''');  
CLPCMD(commande:%size(commande));  
/end-free
```

- Récupérer le résultat dans la variable REP d'un programme RPG via l'API QREXQ qui permet d'écrire et lire les données entre REXX et d'autres langages, via une data queue.

```
*- Déclaration des variables  
D LNG          S          5  0  
D REP          S          256  
D WMICf5       S          256  
D WMICf6       S           1  
D WMIPGM       S          10  
DBfREX1        DS              INZ  
D WMIBf1       1          4B  0  
D WMIBf2       5          8B  0  
D WMIBf3       7         10B  0  
C*- Lire la queue  
C              MOVEL      'QREXQ'      WMIPGM  
C              MOVEL      *BLANK       WMICf5  
C              MOVEL      'P'          WMICf6  
C              Z-ADD      256          WMIBf1  
C              Z-ADD      0            WMIBf2  
C              Z-ADD      0            WMIBf3  
C              CALL      WMIPGM  
C              PARM              WMICf6  
C              PARM              WMICf5  
C              PARM              WMIBf1  
C              PARM              WMIBf2  
C              PARM              WMIBf3  
C              MOVEL      *BLANK       REP  
C              MOVEL      WMICf5       REP
```

- En résumé : appel de la procédure REXX avec en paramètre l'expression mathématique à évaluer (ici '125 + 69 \* (12 / 3)'), puis lecture de la data queue via QREXQ pour récupérer le résultat calculé par la procédure REXX.
- Le résultat obtenu dans REP doit normalement être : 401

## Que faire en plus ?

- Paramétrer le passage de l'expression mathématique à évaluer dans une variable pour rendre dynamique l'appel de la procédure REXX.
- Tester le 5ème paramètre (WMIB£3) qui représente le code retour (succès ou erreur).
- Plein d'autres choses.

## Quelle utilisation ?

- Vous mettez en place une saisie de paramétrage pour des calculs.

## Appel d'une procédure REXX à partir d'un programme RPG

---

- Par exemple, l'utilisateur peut saisir dans un fichier : QTE\_CMD \* PRIX \* CAT\_PROF1 ou QTE\_CMD \* PRIX \* CAT\_PROF2.
  - QTE\_CMD représente une variable contenant la quantité commandée pour un article, PRIX représente une variable contenant le prix de l'article commandé, CAT\_PROF1 représente le taux de réduction appliqué à la catégorie professionnelle 1, et CAT\_PROF2 représente le taux de réduction appliqué à la catégorie professionnelle 2.
- Lors de la lecture du paramétrage, vous obtenez la chaîne de caractère 'QTE\_CMD \* PRIX \* CAT\_PROF1' :
- Remplacer dans la chaîne de caractère 'QTE\_CMD \* PRIX \* CAT\_PROF1' QTE\_CMD par sa valeur en numérique, idem pour PRIX et CAT\_PROF1. Pour obtenir par exemple : '3 \* 15.7 \* 0.90'.
  - Passer la chaîne en paramètre lors de l'appel de la procédure REXX.
  - Vous obtenez en retour l'expression calculée.
- Lors de la lecture du paramétrage, vous obtenez la chaîne de caractère 'QTE\_CMD \* PRIX \* CAT\_PROF2' :
- Remplacer dans la chaîne de caractère 'QTE\_CMD \* PRIX \* CAT\_PROF2' QTE\_CMD par sa valeur en numérique, idem pour PRIX et CAT\_PROF2. Pour obtenir par exemple : '3 \* 15.7 \* 0.85'.
  - Passer la chaîne en paramètre lors de l'appel de la procédure REXX.
  - Vous obtenez en retour l'expression calculée.

## Conclusion

**La puissance de la solution, est que vous pouvez saisir une expression mathématique complexe : opérateurs (\*, +, -, /, ...), parenthèses gauche et droite. Du moment que la chaîne représente une opération mathématique correcte, cela fonctionne. Une solution souple et efficace de paramétrage.**

Attention si vous appelez la procédure à répétition dans des traitements de masse, car il y a tout de même un fort impacte sur les performances.

*Post-scriptum :*

*Voir aussi les articles :*

- [REXX400](#)
- [Calculatrice en REXX](#)