



Extrait du Environnement iSeries

<https://xdocs400.com/spip.php?article237>

# REXX400

- Les articles -



Date de mise en ligne : dimanche 6 mars 2005

## **Description :**

Exemples d'utilisation de REXX sur AS400

---

**Environnement iSeries**

---

## Présentation

- Le langage de programmation REXX est présent sur la plupart des systèmes d'exploitation.
- Langage structuré, simple à apprendre, il offre un jeu d'instruction complet pour la programmation.
- REXX est disponible sur tous les AS400.

```

CMDREX                               Commandes REXX

Choisissez une des options :

Commandes
  1. Add REXX Buffer                   ADDRBUF
  2. Remove REXX Buffer                RMVREXBUF
  3. Start REXX Procedure              STRREXPRC
  4. Trace REXX                       TRCREX

```

## Commandes concernant REX400

### Caractéristique du REXX

- Langage interprété :
  - &mdash; Le REXX n'est pas compilé pour produire un programme exécutable, il est simplement interprété au moment de l'exécution.
  - &mdash; Sur AS400 il s'agira donc d'un membre de fichier source (par exemple QREXSRC.EXEMPLE).
- Langage indépendant du système d'exploitation :
  - &mdash; Le REXX est indépendant en grande partie du système d'exploitation sur lequel il s'exécute.
  - &mdash; Il permet de lancer des commandes en les insérant comme une chaîne de caractère. Si cette chaîne ne correspond pas à une syntaxe REXX valable, elle sera passée à l'interpréteur de commande.

```

***** Début des données
0001.00 /* */
0002.00 do forever
0003.00   'ping 192.120.125.21'
0003.01   call SysSleep 300
0004.00 end
***** Fin des données

```

## PING

La chaîne 'ping 192.120.125.21' n'est pas une instruction REXX, elle est simplement passée à l'interpréteur de commande. Si cette commande s'avérait incorrecte c'est l'interpréteur de commande qui renverrait une erreur (et non pas REXX).

- Un langage au format libre :
  - &mdash; Une instruction simple peut enjambrer beaucoup de lignes ou des instructions multiples peuvent être écrites sur une ligne simple.
  - &mdash; Les instructions peuvent commencer dans n'importe quelle colonne.
  - &mdash; Des espaces ou des lignes entières peuvent être sautés. Pas de distinction majuscule, minuscule, ou mélangé. REXX n'exige pas la numérotation de ligne.
- Pas de typage de variable :
  - &mdash; REXX considère toutes les données comme des chaînes de caractères.
  - &mdash; REXX n'exige pas que des variables ou les tableaux soient déclarés comme chaînes de caractères ou des nombres. REXX exécutera l'arithmétique sur n'importe quelle chaîne de caractères qui représente un nombre valide,

y compris ceux dans des formats exponentiels.

```
MAVAR = '40'
SAY MAVAR + 9
```

Affichera 49

- Une variable non-initialisée contient son propre nom "SAY TOTO" affiche TOTO si aucune valeur n'a été attribué à la variable TOTO.

### Exemple d'utilisation

- Un CL CALLREX reçoit en paramètre un expression à évaluer (par exemple « (435 \* 45) / 2 »
- Un source REXX EVALEXP interprète cette expression et retourne le résultat (via la LDA). Cet exemple utilise une particularité intéressante du REXX qui permet d'interpréter une expression.

```
0001.00 /*
0002.00 /* PROGRAMME : EVALUATION D'UNE EXPRESSION
0003.00 /* CREE PAR : Serge GOMES
0004.00 /*-----
0005.00          PGM          PARM(&ENTREE)
0006.00          DCL          VAR(&ENTREE) TYPE(*CHAR) LEN(1000)
0007.00          DCL          VAR(&BLANC) TYPE(*CHAR) LEN(128)
0008.00          CHGDTAARA  DTAARA(*LDA (201 128)) VALUE(&BLANC)
0009.00
0010.00          STAREXPRC  SRCMBR(EVALEXP) SRCFILE(MABIB/QREXSRC) +
0011.00                      PARM(&ENTREE)
0012.00          MONMSG      MSGID(CPF0000)
0013.00          RTVDTAARA  DTAARA(*LDA (201 128)) RTNVAR(&PHRASE)
0014.00
0015.00 /*          POUR DEMONSTRATION
0016.00          DSPDTAARA  DTAARA(*LDA)
0017.00 ENDPGM
```

#### Source du CL CALLREX

On suppose ici que l'expression (&ENTREE) est valide et que notre programme REXX est dans la bibliothèque MABIB et dans le fichier source QREXSRC Le CL utilise la position 201 de la LDA (DTAARA) pour dialoguer avec REXX

```
0001.00  PARSE ARG expr
0002.00  position = POS(" ", expr)
0003.00  lng = position - 1
0004.00  express = Left(expr, lng)
0005.00  INTERPRET "result =" express
0006.00  sortie:
0007.00  /* AS400 */
0008.00  retour = result
0009.00  'CHGDTAARA DTAARA(*LDA (201 328)) VALUE(''retour'')'
0010.00  EXIT
```

#### Source REXX EVALEXP