



Extrait du Environnement iSeries

<http://xdocs400.com/spip.php?article240>

# Pourquoi utiliser le RPG free-form ?

- Les articles -



Date de mise en ligne : mardi 26 avril 2005

## **Description :**

Une liste des avantages que le « free-form » a, par rapport au style traditionnel fixed-format.

---

**Environnement iSeries**

---

**Le style de codage free-form est disponible pour RPG IV depuis la version V5R1 du OS/400 (printemps 2001). La syntaxe free-form est une grande évolution pour le langage RPG et elle a beaucoup d'avantages en terme d'efficience. Les développeurs ne sont enfin plus liés aux colonnes fixes du langage RPG. Maintenant on peut tout simplement saisir la logique RPG dans un mode naturel de gauche à droite, similaire au style de codage des autres langages évolués (comme Java ou Visual Basic).**

**La syntaxe free-form est permise là où les spécifications de la carte C sont rentrées normalement.**

**Quels sont les avantages que le « free-form » (forme libre) a par, rapport au style traditionnel fixed-format (format fixe). C'est le sujet de cet article.**

[1]

## Sur la base des articles :

"Why Move to Free-Form RPG ?" de [linomasoftware](#).

"V5R1 Enhancements to RPG IV" de [Barbara Morris](#).

## Avantages

1. La logique de type imbriqué peut être indentée.
2. Le code source peut être saisi plus rapidement
3. Il y a plus de place pour saisir des expressions longues
4. Les commentaires (qu'on met à droite des opérations) peuvent suivre tout de suite après le code
5. Le format free peut cohabiter avec le format fixe.
6. Un nombre de plus en plus important des exemples est disponible uniquement en free-form
7. IBM concentre ses ressources sur le RPG free-form
8. Plus facile à apprendre pour les nouveaux venus
9. Enrichit votre expérience

## Avantages détaillés.

Les programmes en free code ont la source du même type et se compilent de la même façon que les programmes RPG IV en format fixe (le compilateur RPG de IBM est assez intelligent pour saisir la différence entre les deux styles).

**La logique imbriquée peut être indentée**

## Pourquoi utiliser le RPG free-form ?

---

Comprendre et mettre à jour la logique imbriquée peut être très difficile dans le RPG format fixe, surtout quand la logique s'étend sur plusieurs niveaux. Essayer de trouver quelle logique est associée aux IF, ELSE et DO peut consommer beaucoup de temps. Par exemple, essayez de comprendre rapidement la logique de l'exemple suivant en format fixe :

```
C      If Test = 3
C      Eval Work = A + B
C      If Work = 1
C      Eval A = A + 1
C      Else
C      If Work = 2
C      Eval B = B + 1
C      EndIf
C      EndIf
C      Eval C = C - 1
C      Else
C      Return
C      EndIf
```

La logique ci-dessus est difficile à suivre, même si le niveau d'imbrication est uniquement 3. On est tenté d'imprimer l'exemple et utiliser un crayon pour tracer des lignes entre IF, ELSE et ENDIF pour mieux comprendre.

Voici, pour comparaison, le codage de la même logique en format free. Cela rend le source effectivement plus lisible.

```
If Test = 3;
  Work = A + B;
  If Work = 1;
    A = A + 1;
  Else;
    If Work = 2;
      B = B + 1;
    EndIf;
  EndIf;
  C = C - 1;
  Else;
    Return;
  EndIf;
```

**Le code source peut être saisi plus rapidement.**

Au lieu de se concentrer sur la saisie des opérations et des variables dans les colonnes correctes, le style free-form permet de saisir le code naturellement, de gauche vers la droite. Il faut tout simplement insérer une ligne et y saisir la logique.

Comme bonus supplémentaire, il n'est pas nécessaire de saisir EVAL ou CALLP quand on attribue des valeurs ou on appelle des procédures. Ces codes d'opérations sont compris par défaut :

Format fixe :

```
C      Eval Work = A + B
C      CallP Process(Work)
```

Format free :

```
Work = A + B;  
Process(Work);
```

Comme exercice, on peut compter le nombre des touches qu'il faut appuyer pour saisir une opération EVAL dans les deux styles. Pour saisir une opération EVAL en utilisant le format fixe par l'éditeur SEU de IBM, le développeur fera les actions suivantes :

1. Saisir IPCX sur un numéro de séquence et appuyer Entrée pour obtenir l'invite (utilise 5 touches)
2. Touche "Tab" pour aller dans la colonne de l'opération (3 fois)
3. Saisir l'opération EVAL dans cette colonne (4 touches)
4. "Tab" pour aller dans la colonne du facteur 2 (1 touche)
5. Saisir l'expression
6. Appuyer sur Entrée (1 fois)

Sans compter les touches utilisées pour saisir l'expression, une simple opération EVAL demande l'utilisation de 14 touches dans le format fixe.

Si on utilise le format free :

1. Saisir un "I" sur un numéro de séquence et appuyer Entrée pour insérer une nouvelle ligne (2 touches)
2. Saisir l'expression sans l'opération EVAL
3. Saisir un ";" à la fin de l'expression (1 touche)
4. Appuyer sur Entrée (1 touche)

De nouveau, sans compter l'expression, cela demande 4 touches pour rentrer la même ligne dans le format free. Cela fait 10 touches de moins que dans le format fixe.

Si on considère les centaines de lignes saisies dans un programme RPG normal, le temps total qu'on gagne avec le free devient important.

### **Plus de place pour saisir des expressions longues.**

Dans le format fixe RPG, il y a 45 caractères disponibles dans la zone du facteur 2 étendu. Si vous avez besoin de saisir une expression plus longue que 45 caractères, vous devez créer une nouvelle ligne de continuation pour saisir le reste. En utilisant le style free-form, vous pouvez saisir des expressions allant jusqu'à 73 caractères sur une même ligne.

Format fixe :

```
Eval          MailLabel = Name + Address1 + Address2 +  
City + State + Zip
```

Format free :

```
MailLabel = Name + Address1 + Address2 + City + State + Zip;
```

Le nombre des lignes nécessaires dans le free-form est plus petit, et les expressions seront plus faciles à lire et à maintenir.

### Les commentaires peuvent suivre immédiatement la logique.

Dans le RPG format fixe, les commentaires faits à la droite du code doivent commencer dans la position 80. Si le fichier source a la longueur de 112 bytes, alors vous avez 20 caractères pour faire un commentaire. Cela vous oblige à faire beaucoup d'abréviations.

Format fixe :

```
C          Eval Work = A + B rpt. total NE+IA
```

Format free :

```
Work = A + B; // report total for Nebraska and Iowa
```

### Free-form peut cohabiter avec les spécifications en format fixe.

Bien qu'il soit recommandé d'avoir toute la logique saisie soit en format fixe, soit en format free, le compilateur RPG permet de mélanger les deux dans un même programme. Il faut simplement saisir les directives de compilation /FREE et /END-FREE et insérer entre celles-ci le code en format free. Tout ce qui est au delà de ces directives de compilation doit être saisi en format fixe.

Exemple :

```
C          MOVEA WORKA WORKB
          /FREE
          A = A + 1;
          B = B - 1;
          /END-FREE
```

Presque toutes les opérations du format fixe ont un équivalent en format free. Uniquement quelques opérations, comme MOVEA, BITON et BITOFF n'ont pas (encore) un équivalent.

### Le nombre des exemples de programmes en free-form augmente

La plupart des programmeurs RPG apprennent une partie des techniques RPG en regardant des exemples de code dans les manuels IBM, des livres et des articles. Comme free-form est déjà disponible depuis plusieurs années, la plupart des auteurs écrivent maintenant leurs livres dans ce style. De plus en plus code RPG disponible pour téléchargement est en free-form aussi. Si vous ne comprenez pas le RPG free-form, il vous sera difficile de profiter de ces nouveaux exemples de code RPG.

### IBM se concentre sur le RPG free-form pour ses évolutions

Il y a une suite d'opérations qui ne sont disponibles que dans free-form. Par exemple, pour retrouver des enregistrements dans un fichier, on n'est plus obligé d'utiliser une KLIST, mais on peut saisir directement les zones de la clé multiple, suivant des opérations comme CHAIN ou READE :

```
/free
// Chain avec les noms des variables
Chain (Zone1 :Zone2) Fichier;
//Chain avec les valeurs
```

## Pourquoi utiliser le RPG free-form ?

```
Chain ('France' : 'Paris') Fichier ;  
/end-free
```

Une évolution disponible uniquement dans free-form est la mise à jour d'un nombre limité des zones d'un fichier :

```
/free  
Update %Fields(Zone1 :Zone2 :Zone3) Fichier ;  
/end-free
```

Il est probable que les prochaines techniques rajoutées par IBM soient disponibles uniquement dans le style free-form.

### Plus facile à apprendre pour les nouveaux venus.

La plupart des universités entraînent leurs étudiants dans des langages free-form comme C, Java ou Visual Basic. Les possesseurs des AS/400 vont alors utiliser de plus en plus le RPG free code, pour attirer les jeunes développeurs (qui ne seraient pas très enthousiastes à programmer dans un langage restrictif à format fixe). La formation dans RPG free-form sera plus attirante et confortable pour les participants, puisque le langage ressemble à ceux qu'ils ont étudié à l'école.

Même pour les programmeurs RPG expérimentés, qui développent aussi en Java, il est plus facile de switcher entre Java et RPG free-form.

### Enrichit votre expérience.

A part la satisfaction personnelle d'apprendre une nouvelle technologie, il est important aussi de savoir que accumuler des nouvelles compétences va vous assurer une carrière stable dans l'informatique. Ne pas mettre à jour vos compétences peut vous mener à une carrière stagnante. Pour le développeurs RPG, le style free-form demandera uniquement quelques journées d'étude avant la bonne maîtrise de celui-ci. C'est un temps bien investi, si l'on tient compte des avantages reçus à l'échange.

Liste des codes d'opérations en format fixe et leurs remplaçants en free-form

Code	Remplaçant free form
ADD	l'opérateur "+"
ADDUR	l'opérateur "+" et les fonctions %YEARS, %DAYS, %MINUTES, etc.
ALLOC	fonction %ALLOC
ANDxx	l'opérateur "AND"
BITOFF, BITON	aucun remplaçant
CABxx	aucun remplaçant
CALL, CALLB	le code opération "CALLP"
CASxx	group "SELECT" avec "EXSR"
CAT	l'opérateur "+"
CHECK	fonction %CHECK

## Pourquoi utiliser le RPG free-form ?

CHECKR	fonction %CHECKR
COMP	code opération "IF"
DEFINE	définitions dans la carte "D"
DIV	l'opérateur "/" ou la fonction %DIV
DO	code opération "FOR"
DOUxx, DOWxx	codes opérations "DOU" et "DOW"
END	ENDIF, ENDDO etc.
EXTRCT	fonction %SUBDT
GOTO	aucun remplaçant
IFxx	IF
LOOKUP	fonctions %LOOKUP, %LOOKUPxx, %TLOOKUP, %TLOOKUPxx
MOVE	EVAL ou EVALR, avec les fonctions %CHAR, %UCS2, %GRAPH, %DATE, %TIME et %TIMESTAMP
MOVEA	EVAL
MOVEL	EVAL avec les fonctions %CHAR, %UCS2, %GRAPH, %DATE, %TIME et %TIMESTAMP
MULT	l'opérateur "**"
MVR	fonction %REM
MxxZO	aucun remplaçant
OCCUR	fonction %OCCUR
ORxx	opérateur "OR"
PARM	définition dans la carte "D"
PLIST	définition dans la carte "D"
REALLOC	fonction %REALLOC
SCAN	fonction %SCAN
SETOFF, SETON	EVAL
SHTDN	fonction %SHTDN
SQRT	fonction %SQRT
SUB	l'opérateur "+"
SUBDUR	l'opérateur "-" avec les fonctions %YEARS, %DAYS, %MINUTES
SUBST	fonction %SUBST
TAG	aucun remplaçant

## Pourquoi utiliser le RPG free-form ?

TESTB	aucun remplaçant
TESTN	aucun remplaçant
TESTZ	aucun remplaçant
TIME	fonctions %DATE, %TIME et %TIMESTAMP
WHENxx	code opération "WHEN"
XFOOT	fonction %XFOOT
XLATE	fonction %XLATE
Z-ADD, Z-SUB	EVAL

Tous les autres codes d'opérations restent inchangés :

- ACQ
- BEGSR
- CALLP
- CHAIN
- CLEAR
- CLOSE
- COMMIT
- DEALLOC
- DELETE
- DOU
- DOW
- DSPLY
- DUMP
- ELSE
- ELSEIF
- ENDDO
- ENDFOR
- ENDIF
- ENDMON
- ENDSL
- ENDSR
- EVAL
- EVALR
- EXCEPT
- EXFMT
- EXSR
- FEOD
- FOR
- FORCE
- IF
- IN
- ITER
- LEAVE
- LEAVESR



- MONITOR
- NEXT
- ON-ERROR
- OPEN
- OTHER
- OUT
- POST
- READ
- READC
- READE
- READP
- READPE
- REL
- RESET
- RETURN
- ROLBK
- SELECT
- SETGT
- SETLL
- SORTA
- TEST
- UNLOCK
- UPDATE
- WHEN
- WRITE

Une description exhaustive des codes d'opérations avec des exemples se trouve dans le document de IBM Avec des exemples en free-form (document IBM en anglais) : [ILE RPG REFERENCE](#). A voir surtout le chapitre 22 ( built-in-functions, ou BIFs, qui remplacent de plus en plus les codes d'opérations traditionnels dans RPG).

---

[1] Auteur : Cris SEFCIUC