



Extrait du Environnement iSeries

<http://xdocs400.com/spip.php?article340>

Débogage des Programmes ILE

- Les articles -



Date de mise en ligne : lundi 14 mai 2007

Description :

L'outil de débogage iseries couvre tous les langages et tous les types de programme de cet environnement.

Environnement iSeries

Les bonnes pratiques pour utiliser l'outil standard de débogage de la plateforme AS400-iSeries-system i, qui couvre tous les langages et tous les types de programme de cet environnement.

Dans une des versions V2 de l'OS/400 (je ne me souviens plus laquelle), IBM a intégré ISDB (Interactive Source Debugger) que l'on pouvait utiliser avec les vieux programmes RPG pour visualiser le source à l'écran. ISDB est un bel outil mais il ne supporte pas tous les langages disponibles sur le système i. Il ne fonctionne pas pour les programmes ILE ni pour Java, mais seulement pour les programmes OPM CL, RPG et COBOL.

A partir de la V3 de l'OS/400, IBM a amélioré la commande STRDBG pour en faire un débogueur interactif. A l'inverse de ISDB, cette commande fonctionne avec tous les langages disponibles sur le système i. Cela inclus tous les langages ILE et OPM ainsi que Java. De plus cette commande permet de déboguer tous ces types de source en même temps.

Par exemple, si un programme CL OPM appelle un programme ILE et que cet ILE est écrit en RPG IV avec des appels de modules écrits en C ou en COBOL, vous pouvez suivre le déroulement du programme dans ces différents sources en utilisant la touche fonction F22. Quelle souplesse !

Pour permettre le débogage interactif de vos programmes, il faut indiquer une vue de débogage (DBGVIEW) lors de leur compilation. Tous les compilateurs ILE ont une option DBGVIEW qui peut prendre les valeurs suivantes :

DBGVIEW(*STMT) = Par défaut, le compilateur n'intègre dans l'objet que les numéros de ligne des instructions (Statement) compilées. Avec cette option, vous n'obtiendrez pas de débogage interactif.

DBGVIEW(*SOURCE) = Cette option permet au débogueur d'afficher le membre source de votre programme. Ce résultat ressemble à ce que le débogueur produit d'habitude. Le problème de cette option est que le source n'est pas intégré dans l'objet, par conséquent si le membre source est modifié sans être recompilé, le débogueur affichera des informations erronées. De même si votre objet programme est déplacé sur un autre système, le source peut ne plus être disponible, ce qui empêchera le débogage.

DBGVIEW(*LIST) = Une copie du listing de compilation est incorporée dans l'objet. Quand le débogueur est activé ; il utilise ce listing au lieu du membre source, cela garantit que le programme sera débogué correctement même si le membre source est modifié ou si l'objet change de système. C'est l'option qui est recommandée.

DBGVIEW(*ALL) = Permet toutes les options précédentes au moment du débogage.

Evidemment, l'inconvénient majeur de l'option DBGVIEW(*LIST) est qu'elle nécessite plus d'espace disque pour stocker les objets de programmes puisqu'ils contiennent également le source. Mais sur tous les systèmes i en production que j'ai rencontré, la plus grosse partie du disque est toujours la base de données. Le code n'occupe qu'une part négligeable, et donc augmenter d'un faible pourcentage cette part déjà réduite n'a pas un gros impact.

Même si l'espace disque a de l'importance pour vous, rappelez-vous que les disques ne sont pas très chers comparés à la perte de rentabilité. Si un programmeur perd plus de 15 heures par an à rechercher des sources qui

Débogage des Programmes ILE

ne sont plus en phase avec ses objets, cela vous coûtera plus cher que si vous aviez installé un disque supplémentaire.

Si je n'ai pas réussi à vous convaincre d'utiliser l'option `DBGVIEW(*LIST)`, vous pouvez utiliser l'option `DBGVIEW(*SOURCE)` qui vous permettra d'utiliser la commande `STRDBG` en interactif sans consommer d'espace disque supplémentaire.

Les programmes OPM (classiques) peuvent également être débogués interactivement avec `STRDBG`. Il faut les compiler avec l'option `OPTION(*SRCDBG)` ou encore `OPTION(*LSTDBG)` pour vous permettre de les déboguer. Ces options sont semblables à `DBGVIEW(*SOURCE)` et `DBGVIEW(*LIST)`, respectivement.

Une fois que vous avez compilé votre programme avec l'option appropriée, vous pouvez démarrer le débogueur en tapant :

```
STRDBG PGM(monProgramme)
```

Ou bien, pour un programme OPM :

```
STRDBG PGM(monProgramme) OPMSRC(*YES)
```

Le débogueur met en place l'environnement de débogage et affiche le source de votre programme à l'écran. Voici un exemple de ce que vous pourriez avoir à l'écran :

```
=====
```

Source d'un module

```
Programme : XLCRTDEMO   Biblio : TIPSNL   Module : XLCRTDEMO
```

```
16  001500
```

```
17  001600 FDIVSALES IF E      K DISK
```

```
18  001700
```

```
19  001800 /copy qrpglesrc,hssf_h
```

```
20  001900
```

```
21  002000 D CreateCellStyles...
```

```
22  002100 D          PR
```

- 23 002200

- 24 002300 D AddMonth PR

- 25 002400 D book like(HSSFWorkbook)

- 26 002500 D month 2P 0 value

- 27 002600 D monthname 20A varying const

- 28 002700

- 29 002800 D FormatColumns PR

- 30 002900 D sheet like(HSSFSheet)

A Suivre..

Débogage . . .

F3=Arrêter programme F6=Ajouter/Effacer pt arrêt F10=Pas à pas F11=Variable F12=Reprendre F17=Guet de variable F24=Autres touches

=====

Vous pouvez paginer dans le source et insérer des points d'arrêt là où vous voulez interrompre le déroulement du programme. Utilisez les touches de fonction F10 ou F22 (Pas à pas) pour interrompre le programme à la fin de l'instruction suivante. La différence entre ces deux touches de fonction est apparente lorsque le programme appelle un autre module ou une fonction ou encore une procédure. Si vous utilisez la touche F22, le débogage se poursuit dans les instructions de la routine appelée, alors que si la touche fonction F10 est utilisée, vous ne verrez pas les instructions de la routine appelée, l'exécution s'interrompt après le retour de cette routine.

La touche fonction F14 est utilisée pour basculer d'un module à l'autre dans le programme ou pour ajouter des programmes ou des modules dans l'environnement de débogage. Cela permet d'étendre le débogage et d'insérer de nouveaux points d'arrêt.

Une fois les points d'arrêt insérés, appuyer sur F12 pour reprendre l'exécution. Vous quitterez alors l'affichage interactif pour revenir à la ligne de commande. A la différence d'ISDB, le débogueur ne lance pas l'application. Il vous faut lancer le programme au moyen de la commande CALL ou d'une option de menu mais impérativement dans l'espace de travail où vous venez d'effectuer le STRDBG.

Lorsque vous aurez fini de déboguer, utilisez la commande ENDDBG qui supprime l'environnement de débogage. De même si vous devez recompiler votre programme assurez-vous de quitter le mode débogage (ENDDBG) et de le redémarrer (STRDBG) pour que l'environnement se reconstitue avec la nouvelle version de l'objet compilé.

Bon débogage !