



Extrait du Environnement iSeries

<http://xdocs400.com/spip.php?article262>

La magie de la clé libre

- Les articles -



Date de mise en ligne : jeudi 22 décembre 2005

Description :

Une technique de programmation simple, rapide et efficace, qui intéressera aussi bien les développeurs Adelia que RPG. Si vous voulez en savoir plus, lisez la suite.

Environnement iSeries

Je vais vous présenter ci-dessous une technique de programmation toute simple, mais qui m'a rendu, et continue de me rendre de grands services. Cette technique m'avait été recommandée par un expert du développement sur plateforme iSeries, Frédéric Ancel (que j'en profite pour saluer ici). Elle s'applique tout aussi bien à des développements Adelia que RPG.

Je vais vous présenter ci-dessous une technique de programmation toute simple, mais qui m'a rendu, et continue de me rendre de grands services. Cette technique m'avait été recommandée par un expert du développement sur plateforme iSeries, Frédéric Ancel (que j'en profite pour saluer ici). Elle s'applique tout aussi bien à des développements Adelia que RPG.

Prenons un exemple.

Vous devez développer un programme d'édition de bons de préparation (BP). Vous en faites une première version, adaptée aux besoins d'un premier client, et tout va bien. D'ailleurs, profitons-en pour rentrer un peu dans la technique. Votre traitement est constitué d'un programme CL, qui pilote différentes opérations, telles que :

- 1 - montage d'un fichier de travail dans QTEMP
- 2 - appel d'un programme d'alimentation du fichier de travail (stockage de la liste des commandes dont le BP doit être édité)
- 3 - appel d'un programme d'édition des bons de préparation spécifique à la première société

Quelques temps plus tard, vous devez personnaliser ce programme pour une autre société, et vous vous apercevez que la nouvelle société souhaite que ses BP soient triés sur des critères qui ne sont pas ceux de la première société (par exemple, la première société souhaite que ses BP soient triés par points de livraison, tandis que l'autre souhaite que ses BP soient triés par tournées). Vous retrouvez vos manches, en vous disant : "qu'à cela ne tienne, je vais ajouter un autre fichier logique sur le fichier de travail, et j'éditerai les BP tels que demandé par le nouveau client). Mais soudain, une angoisse vous saisit, et vous vous dites : "et la prochaine société, comment voudra-t-elle que ses BP soient édités ?" Je rencontre cette problématique fréquemment, puisqu'au moment où j'écris cet article, j'ai en charge un portefeuille de 11 sociétés. Je n'ai certes pas 11 tris différents pour l'édition des BP, mais j'en ai actuellement 5, et il est probable que je doive en créer de nouveaux dans l'avenir. Je retrouve bien évidemment le même problème sur d'autres éditions, telles que "bons de livraison", "factures", etc...

Et c'est là que la technique de la clé libre intervient.

En conservant la technique du fichier de travail déjà évoquée, j'ajoute à ce fichier une zone que j'appelle "clé libre" et que je taille généralement à 100 caractères (pour être tranquille). Je prévois un fichier logique sur mon fichier de travail, exploitant dans son index cette "clé libre" et uniquement elle.

Il ne reste plus qu'à alimenter cette "clé libre" par concaténation, en fonction des critères de tris demandés par telle ou telle société. Je vous recommande d'ailleurs de prévoir dès le départ un fichier de paramétrage, vous permettant, pour chaque société, de définir le critère de tri souhaité, ainsi que le programme d'édition final.

Dans votre programme d'alimentation du fichier de travail (qui sera le même pour toutes les sociétés), vous aurez un bout de code de ce type (dans mon exemple écrit en Adelia, la zone clé libre de mon fichier de travail s'appelle

KW_CLE_LIB) :

```
KW_CLE_LIB = *BLANK
SI W_TRI = "01"
*-- Tri de la société 1 (par numéro de commande)
KW_CLE_LIB = KW_NUM_CDE
SINON
SI W_TRI = "02"
*-- Tri de la société 2 (par client et numéro de commande)
KW_CLE_LIB = KW_NUM_CLI // KW_NUM_CDE
SINON
SI W_TRI = "03"
*-- Tri de la société 3 (par date de livraison, tournée de commande et numéro de commande)
KW_CLE_LIB = KW_DAT_LIV // KW_NUM_TOU // KW_NUM_CDE
SINON
etc...
FIN
FIN
FIN
```

Bien évidemment, il faut s'assurer que les ruptures (s'il y en a) dans le programme d'édition des BP final, sont conformes au tri défini dans la clé libre. Pour faciliter d'ailleurs la détermination des ruptures dans le programme d'édition, il est vital que le fichier de travail contienne à la fois la clé libre, et les zones ayant servi à alimenter cette même clé libre (dans l'exemple ci-dessus, ce seraient les zones KW_NUM_CDE, KW_NUM_CLI, etc...). Il serait en effet dommage de devoir redécouper la clé libre (au moyen d'une DS ou de tout autre moyen), pour pouvoir détecter les ruptures.

En Conclusion :

Cette technique est vraiment très pratique. Ses détracteurs pourraient lui opposer cependant 2 arguments :

- la concaténation de chaînes est gourmande en ressources => bof, si votre AS/400 n'est pas suffisamment puissant pour ce type d'opération, peut être devriez-vous prévoir de l'upgrader rapidement, isn't it ?
- il aurait semblé plus simple d'ajouter des fichiers logiques au fichier de travail : oui mais dans ce cas, si votre fichier de travail est constitué d'une dizaine de fichiers logiques (1 fichier logique par tri), le programme CL va monter le fichier dans QTEMP et ses 10 fichiers logiques, pour éditer des BP qui eux n'exploitent qu'un seul type de tri au sein d'une société. De plus, le "montage" de fichier dans QTEMP n'est pas instantané, il consomme du temps machine. On pourrait pallier ce problème en ne montant dans QTEMP que le fichier logique nécessaire pour une société donnée. Il me semble que ce serait compliquer inutilement, et dangereusement, ce pauvre programme CL (enfin, bon, si vous avez du temps à perdre...).

Post-scriptum :

Cette technique de programmation toute simple vous séduira si vous décidez de l'essayer. Encore merci à Frédéric Ancel pour ce bon conseil, et pour tous les autres.