

# Envoi des spools IBM-i vers MAPPING ONYX Windows ou Linux

## **Création d'une RemoteOUTQ sur iSeries pour envoi des spools vers M-PS Windows/Linux**

```
CRTOUTQ OUTQ(QGPL/TOWINDOWS) RMTSYS(*INTNETADR) RMTprtQ(FROMISERIES)
AUTOSTRWTR(1) DESTTYPE(*OTHER) TRANSFORM(*NO) INTNETADR(XXX.XXX.XXX.XXX)
DESTOPT('XAIX') DSPDTA(*YES) AUTCHK(*DTAAUT) AUT(*ALL)
```

OUTQ(QGPL/TOWINDOWS) > nom de l'OUTQ à créer, dans cet exemple l'OUTQ TOWINDOWS dans la bibliothèque QGPL

RMTprtQ(FROMISERIES) > nom de la file d'attente créée dans M-PS sur Windows/Linux

AUTOSTRWTR(1) > nombre de Writer démarré automatiquement, un éditeur doit être démarré pour envoyer les spools à distance

TRANSFORM(\*NO) et DESTOPT('XAIX') > paramètres obligatoires

INTNETADR(192.168.216.24) > adresse IP du serveur M-PS Windows/Linux

Commande CHGOUTQ pour modifier une OUTQ existante (pour modifier, s'assurer que l'éditeur est arrêté)

ATTENTION : La valeur du paramètre MAP\_HOST envoyé avec le spool peut être différente selon que le nom DNS est renseigné ou non sur l'AS400. MAP\_HOST = nom DNS de l'AS400 si il existe, sinon il prend la valeur de \*INTNETADR.

## **Envoi de travaux d'impression IBM i vers ONYX par LPR**

Le principe de cette fonctionnalité est d'utiliser la commande LPR native OS/400 pour l'envoi de fichier spool IBMI vers Onyx.

## **Problématique rencontrée avec LPR**

La plus grosse problématique avec l'utilisation d'une RMTOUTQ utilisant le protocole LPR pour l'envoi des spools AS/400 vers Onyx Linux/Windows est la perte des métadonnées car les attributs de spools ne sont pas transportés.

## **Solution**

Avec la commande LPR, il est possible de faire transiter des métadatas grâce à l'utilisation du paramètre **destopt** de la commande LPR.

En l'utilisant comme dans l'exemple ci-dessous à partir du robot Mapping, il est possible de transporter les métadonnées que vous souhaitez :

```
LPR RMTSYS(xxxxx) (...) destopt('Z:map_userdata:KEY=VALUE')
```

Avec :

- "Z:map\_userdata:" : obligatoire pour indiquer à Onyx qu'il s'agit d'une métadonnée
- "KEY" : le mot clé de votre choix
- "VALUE" : la valeur de votre choix

Sous Onyx (Windows ou Linux), les informations se retrouvent alors dans le job sous forme de Mapping User Data :

```
USERDATAS_KEY=VALUE
```

Exemple d'utilisation :

```
LPR RMTSYS(xxxxx) (...)  
destopt('Z:map_userdata:FROMAS400=&USER;&OUTORI;&SPLFNAME;&STARTPAGE;&ENDPAGE  
;&USRDTA;&FORMTYPE')
```

## Problématiques connues

### Longueur du paramètre destopt

Le champs destopt a une longueur maximale de 128 caractères. Dans le cas où cette valeur dépasse les 128 caractères, la commande LPR tombera en erreur. Vous obtiendrez alors le message **Erreurs trouvées dans la commande** dans les logs du robot MAPPING.

### Incompatibilité avec les espaces

Il faut s'assurer qu'il n'y ait pas d'espace (un caractère blank) dans la chaine destopt car son contenu sera verra tronqué après ce caractère blanc. En effet, Onyx recevra uniquement le contenu du destopt jusqu'au premier blanc trouvé.

Pour éviter cela, il faut s'assurer que le contenu du destopt ne contienne pas de "blank"

Il faut donc faire attention au contenu des paramètres utilisées pour le contenu du destopt.

### Paramètres avec des espaces potentiels

Quand vous souhaitez utiliser les paramètres suivants dans le paramètre destopt, il faut faire attention aux points suivants.

Il est également possible de faire un trim droit de certains paramètres afin de supprimer les espaces non voulu à la fin de chacuns d'eux en utilisant

l'opération **&TRIM** (Disponible à partir de la version 10.2.8 d'OPALE)

### Numéro du spool

```
&SPLNUM = '      1'
&CLSPLNUM5 = '000001'
```

### Référence utilisateur

Le paramètre &USRDTA a toujours une longueur de 10 caractères. &USRDTA =  
'REFUSER'  
&SPLUSRDTA = 'REFUSER'

Vous pouvez utiliser le paramètre &TRIM pour supprimer les espaces de fin. Il est possible de le faire uniquement avec le paramètre &SPLUSRDTA (Et non &USRDTA)

```
&TRIM&SPLUSRDTA = 'REFUSER'
```

### Type d'imprimé

Le type d'imprimé a une longueur de 10 caractères. De ce fait, il faudra faire en sorte de supprimer les espaces de fin si vous souhaitez l'utiliser comme valeur du destopt :

```
&FORMTYPE = '*NONE'
&TRIM&FORMTYPE = '*NONE'
```

### Utilisation de l'opérateur &TRIM et &NOTRIM

Comme vu précédemment dans cette présente documentation, il est possible de faire un trim droit de certains paramètres afin de s'assurer que le contenu suivant de soit pas tronqué. Pour ce faire, il faut avoir mettre l'opérataion &TRIM avant le nom du paramètre à trimmer. Celui-ci devra être positionner dans la commande LPR au niveau du destopt avant le ou les paramètres à trimer.

Exemple :

```
LPR RMTSYS('192.168.217.224') PRTQ(PRINTER1) FILE(&SPLF)
JOB(&JOBNUM/&USER/&JOB)
SPLNBR(&SPLNUM)DESTOPT('Z:map_userdata:FROMAS400=&SPLF;&OUTORI;&STARTPAGE;&EN
DPAGE;&TRIM&SPLUSRDTA;&SPLF;&USRDFNDDTA;&FORMTYPE')
```

En effet, tous les paramètres situés après le paramètre &TRIM se trouveront trimé.

Remarque 1 : Si vous souhaitez trimer tous les paramètres, vous pouvez mettre le paramètre &TRIM tout au début de la chaine destopt :

```
LPR RMTSYS('192.168.217.224') PRTQ(PRINTER1) FILE(&SPLF)
JOB(&JOBNUM/&USER/&JOB) SPLNBR(&SPLNUM)
```

```
DESTOPT('Z:map_userdata:FROMAS400=&TRIM&SPLF;&OUTORI;&STARTPAGE;&ENDPAGE;&SPL  
USRDTA;&SPLF;&USRDFNDA;&FORMTYPE')
```

En faisant cela, vous vous assurez du trim de tous les paramètres envoyés à Onyx.

Remarque 2 : Il est possible à tout moment de désactiver le trim du &TRIM en utilisant le paramètre &NOTRIM

```
LPR RMTSYS('192.168.217.224') PRTQ(PRINTER1) FILE(&SPLF)  
JOB(&JOBNUM/&USER/&JOB) SPLNBR(&SPLNUM)  
DESTOPT('Z:map_userdata:FROMAS400=&TRIM&SPLF;&OUTORI;&STARTPAGE;&ENDPAGE;&SPL  
USRDTA;&SPLF;&NOTRIM&USRDFNDA;&FORMTYPE')
```

Dans cet exemple, les paramètres (&SPLF;&OUTORI;&STARTPAGE;&ENDPAGE;&SPLUSRDTA;&SPLF) seront bien trimés mais les deux derniers paramètres (&USRDFNDA;&FORMTYPE) sont quant à eux non pris en compte par le &TRIM.

## Interprétation côté Onyx

Avec cet exemple, il suffit ensuite de découper la chaîne de caractères obtenue directement dans un workflow Mapping, à l'aide de la boîte "String/Couper par champ", en précisant le délimiteur ainsi que le numéro de champ à extraire. Par contre, la position de l'élément est importante dans ce cas.

Une autre solution pourrait consister à passer également le nom des paramètres et utiliser la boîte "String/Couper par séparateur".

Exemple d'utilisation :

```
LPR RMTSYS(XXXXX) (...)  
destopt('Z:map_userdata:FROMAS400=USER=&USER;OUTQ=&OUTORI;SPLFNAME=&SPLFNAME;  
STARTPAGE=&STARTPAGE;STOPPAGE=&ENDPAGE;USRDTA=&USRDTA;FORMTYPE=&FORMTYPE')
```

Ensuite, dans le workflow Onyx, pour récupérer la valeur "USER" par exemple, il faut mettre "USER=" dans le séparateur de début de sous chaîne et ";" dans le séparateur de fin de sous chaîne.



## Set your param

Le paramètre est

USER



Sa valeur est

Command : cutseparator



title

Cut string with separator

Note

icone



Selection icone.

String



Couper par  
position



Couper par  
séparateur



Couper par  
champ



Basename  
du chemin



Dirname du  
chemin



Remplacer  
xyz



Supprimer  
les espaces



Mettre en  
majuscules



Mettre en  
minuscules



Capitaliser



Inverser la  
casse

Standard

Error

Chaine à découper

FROMAS400



Valeur par défaut si aucune valeur n'est trouvée

not found

Séparateur de début de sous-chaine

USER=

Séparateur de fin de sous-chaine

;