

Mettre à jour les ressources NIM

Aaron HONORE

Cet article a pour but de vous aider à mettre à jour les principales ressources de votre environnement NIM. Vous y trouverez également une commande permettant de vérifier le niveau des filesets de ces ressources..

Les ressources pouvant être mises à jour

Les types

Les deux principaux types de ressources pouvant être mises à jour sont :

- lpp_source
- spot

Les listes

- Pour lister l'ensemble des ressources de l'un de ces types, passer les commandes respectives :

- `/> lsrim -t lpp_source`

lpp_source_431	resources	lpp_source
lpp_source_432	resources	lpp_source
lpp_source_421	resources	lpp_source
lpp_source_420	resources	lpp_source
lpp_source_433	resources	lpp_source
lpp_source_510	resources	lpp_source
lpp_source_433_ML10	resources	lpp_source
lpp_source_510_ML02	resources	lpp_source

et

- **/> lsrim -t spot**

```
spot_433  resources  spot
spot_432  resources  spot
spot_431  resources  spot
spot_421  resources  spot
spot_510  resources  spot
```

- Pour lister les informations relatives à une ressource particulière, d'un type donné, passer respectivement les commandes :

- **/> lsrim -l lpp_source_name**

```
$ lsrim -l lpp_source_433
lpp_source_433:
class      = resources
type       = lpp_source
arch       = power
Rstate     = ready for use
prev_state = verification is being performed
location   = /export/lpp_source/lpp_source_433
simages    = yes
alloc_count = 2
server     = master
```

et

- **/> lsrim -l spot_name**

```
$ lsrim -l spot_510
spot_510:
class      = resources
type       = spot
plat_defined = chrp
arch       = power
plat_defined = rs6k
plat_defined = rspc
bos_license = yes
Rstate     = ready for use
prev_state = lppchk operation is being performed
location   = /export/spot/spot_510/spot_510/usr
version    = 5
release    = 1
mod        = 0
alloc_count = 1
server     = master
if_supported = chrp.mp ent
if_supported = chrp.up ent
if_supported = rs6k.mp ent
if_supported = rs6k.up ent
if_supported = rspc.mp ent
if_supported = rspc.up ent
if_supported = rspc.up tok
Rstate_result = success
```

Le *lpp_source*

Qu'est-ce qu'un *lpp_source* ?

Le *lpp_source* correspond à un ensemble de *filesets* installables.

Il peut correspondre à :

- l'ensemble du système d'exploitation AIX (créé à partir des CD d'AIX),
- un niveau de maintenance level,
- une PTF,
- des *filesets* correspondant à un produit à installer.

Informations disponibles à propos du *lpp_source*

Examinons le résultat de la commande “*lsnim*” et analysons les informations qu'elle fournit.

```
$ lsnim -l lpp_source_433
lpp_source_433:
class      = resources
type       = lpp_source
arch       = power
Rstate     = ready for use
prev_state = verification is being performed
location   = /export/lpp_source/lpp_source_433
simages    = yes
alloc_count = 2
server     = master
```

- L'attribut “*location*”

Il permet de savoir où se trouvent la “ressource” et les “*filesets*” contenus par notre *lpp_source*.

L'attribut “*server*” permet de déterminer sur quelle machine de l'environnement NIM se trouve notre ressource.

NB : Depuis l'AIX 5.1, les *filesets* ne se trouvent plus directement dans ce répertoire mais dans le sous-répertoire “*installp/ppc*”.

- L'attribut “*simages*”

Il permet de savoir si notre *lpp_source* est installable ou non ; en effet, on peut créer des “*lpp_source*” contenant uniquement des PTF/APAR qui ne seront donc pas “installables”.

Pour vérifier la cohérence de notre *lpp_source*, on peut passer la commande :

nim -o check lpp_source_name

Cette commande va régénérer la table des matières (fichier “.toc”) de notre *lpp_source* et, si celui-ci peut permettre de faire une installation de base, l'attribut “*simages*” sera positionné à “yes”.

- L'attribut “*alloc_count*”

Il indique le nombre de machines auxquelles cette ressource est allouée.

La plupart des opérations effectuées sur les ressources *lpp_source* (ou *spot...*) doivent l'être lorsque ces dernières ne sont pas allouées.

Vous trouverez, au bas de la page, un exemple de *script* (nommé “resalloc ...”) permettant de lister l’ensemble des machines pour lesquelles la ressource est allouée.

Après avoir déterminé à quelle machine la ressource est allouée, il est nécessaire de désallouer cette ressource pour pouvoir continuer la mise à jour.

Vous pouvez désallouer **toutes** les ressources avec la commande suivante :

```
$ nim -o deallocate -a subclass=all <machine_name>
```

Si cette commande vous renvoie un message d’erreur du type :

```
0042-302 m_deallocate:  the state of «machine_name» prevents this operation
                        from succeeding. Use the «reset» operation to correct
                        its state then retry the intended operation.
```

c’est que la machine est sans doute prête pour une installation ou une autre opération nécessitant l’allocation de ressource. Dans ce cas, il faut faire un “reset” de l’état de la machine :

```
$ nim -Fo reset <machine_name>
$ nim -o deallocate -a subclass=all <machine_name>
```

Pour désallouer **un seul** type de ressource utiliser la commande suivante (exemple dans notre cas : un *lpp_source*) :

```
$ nim -o deallocate -a <ressource_type>=<ressource_name> <machine_name>
```

Exemple de *script* “resalloc ...” :

```
#!/bin/ksh

if [[ $# -eq 0 ]]
then
    echo «A parameter is missing : ressource name»
    echo «Usage : resalloc ressource_name»
else if [[ $# -eq 1 ]]
then
    MAC_LIST='lsnim | grep machines | grep -v master | awk {'print $1'}'
    for MAC in $MAC_LIST
    do
        lsnim -l $MAC | grep -p $1
    done
else
    echo «Too much parameter.»
    echo «Usage : resalloc ressource_name»
fi

fi
```

- L'attribut **“server”**

Indique la machine sur laquelle se trouve la ressource (en général : “master” ; mais NIM peut aussi définir des ressources sur un serveur de ressources distant).

Le spot

```
$ lsrim -l spot433
spot433:
class           = resources
type            = spot
Rstate          = prêt à l'emploi
prev_state      = vérification en cours
location        = /export/spot/spot433/usr
version         = 4
release         = 3
mod             = 3
alloc_count     = 0
server          = master
if_supported    = rs6k.mp ent
if_supported    = rs6k.up ent
if_supported    = rspc.mp ent
Rstate_result   = réussite
plat_defined    = chrp
plat_defined    = rs6k
plat_defined    = rspc
```

- L'attribut **“location”**

Indique le chemin de l'emplacement du “spot” ; il doit normalement contenir l'équivalent de “/usr” d'une machine déjà installée. En effet, le spot correspond à une sorte d'image minimale qui permet de faire un *boot* sur une machine cliente.

- Les attributs **“version”**, **“release”** et **“mod”**

Indiquent le niveau d'AIX installé dans le “spot” (cela correspond au résultat de la commande “oslevel” mais ne donne aucune indication quant au niveau de PTF ou ML installées).

- L'attribut **“server”**

Indique l'emplacement de la ressource “spot” (il est possible de définir une ressource sur le *master* alors qu'elle se trouve physiquement sur une machine de l'environnement NIM).

- Les attributs **“if_supported”** et **“plat_supported”**

Permettent de savoir quels sont les environnements matériels (*hardware*) supportés (interface réseau et plate-forme).

Autres ressources utiles

- **bosinst_data**
Permet d'automatiser les installations.
- **image_data**
Contient les informations relatives au système.
Peut être utilisée pour optimiser (*tuning*) le “client” (réduire la taille d'un *filesystem*...)
- **installp_bundle**
Permet de spécifier l'ensemble des *filesets* à installer sur le “client”.
- **script**
Exécuté en fin d'installation afin de configurer (*customize*) le “client”.
NIM exécute toujours un *script* de configuration après l'installation afin :
 - de paramétrer le réseau du client et de l'intégrer dans l'environnement NIM,
 - d'exécuter le *script* de configuration utilisateur.
- **mksysb**
C'est l'image d'un système.
- **resolv_conf**
Ce fichier permet de configurer un DNS sur le “client”.

Mettre à jour votre environnement NIM

Certaines ressources ne nécessitent aucune commande NIM pour que la mise à jour soit prise en compte, par exemple : *bosinst_data*, *image_data*, *script*... Pour ces ressources, il suffit de retrouver le fichier associé et de le modifier.

Les ressources nécessitant des opérations plus spécifiques relatives à NIM sont le “*lpp_source*” et le “*spot*”.

Le “*lpp_source*”

Il est, tout d'abord, nécessaire de savoir où sont copiés les *filesets* ; pour cela, taper les commandes :

```
$ lsrim -l lpp_source433
```

(Noter le champ “location”, par exemple : “/export/lpp_source/lpp_source433”)

```
$ cd /export/lpp_source/lpp_source433
```

```
$ find . -name .toc
```

- Si la commande “`$ find . -name .toc`” renvoie : “`./installp/ppcl.toc`”

On peut, dans ce cas, mettre à jour le “*lpp_source*” par “*smit*” ou en **ligne de commande**.

- Mise à jour du “lpp_source” par “smit” :

\$ smitty bffcreate

- ➔ Renseigner le chemin des *filesets* que l'on souhaite ajouter au “lpp_source”.
- ➔ Appuyer sur <F4> pour sélectionner les *filesets* que l'on veut copier.
- ➔ Dans le champ “*Directory for storing software package*”, mettre le contenu du champ “location” du “lpp_source” (dans notre exemple : “/export/lpp_source/lpp_source433”).

- Mise à jour du “lpp_source” en **ligne de commande** :

\$ bffcreate -d /dev/cd0 -t /export/lpp_source/lpp_source433 -f <liste des filesets>

- Si la commande “\$ find . -name .toc” renvoie : “.toc”

Il suffit, dans ce cas, de copier les *filesets* voulus dans le répertoire courant (celui où se trouve le “.toc”) :

\$ cp /mnt/cdrom/installp/ppc/filesets /export/lpp_source/lpp_source433/

Puis de régénérer une table des matières correcte :

```
$ cd /export/lpp_source/lpp_source433
$ rm .toc
$ inutoc
```

Pour vérifier que le “lpp_source” contient bien les *filesets* que l'on a copiés il suffit de passer la commande NIM :

\$ nim -o showres -a installp_flags= « L » <lpp_source>

```
$ nim -o showres -a installp_flags= « L » lpp_source433
bos.net:bos.net.ate:4.3.3.0::l:T::::N:Asynchronous Terminal Emulator :::
bos.net:bos.net.ipsec.keymgt:4.3.3.0::l:T::::N:IP Security Key Management :::
bos.net:bos.net.ipsec.rte:4.3.3.0::l:T::::N:IP Security :::
bos.net:bos.net.ipsec.websm:4.3.3.0::l:T::::N:IP Security WebSM :::
bos.net:bos.net.ncs:4.3.3.0::l:T::::N:Network Computing System 1.5.1 :::
bos.net:bos.net.nfs.adt:4.3.3.0::l:T::::N:Network File System Development Toolkit :::
bos.net:bos.net.nfs.cachefs:4.3.3.0::l:T::::b:CacheFS File System :::
bos.net:bos.net.nfs.client:4.3.3.0::l:T::::b:Network File System Client :::
bos.net:bos.net.nfs.server:4.3.3.0::l:T::::Y:Network File System Server :::
bos.net:bos.net.nis.client:4.3.3.0::l:T::::Y:Network Information Service Client :::
bos.net:bos.net.nis.server:4.3.3.0::l:T::::Y:Network Information Service Server :::
bos.net:bos.net.nisplus:4.3.3.0::l:T::::N:Network Information Services Plus (NIS+) :::
bos.net:bos.net.ppp:4.3.3.0::l:T::::N:Async Point to Point Protocol :::
bos.net:bos.net.tcp.adt:4.3.3.0::l:T::::N:TCP/IP Application Toolkit :::
bos.net:bos.net.tcp.client:4.3.3.0::l:T::::b:TCP/IP Client Support :::
bos.net:bos.net.tcp.server:4.3.3.0::l:T::::Y:TCP/IP Server :::
```

De la même manière que la commande “installp”, cette commande renvoie le contenu du “lpp_source” mis en forme.

En effet, après lecture du fichier “table des matières”, on peut directement utiliser la commande “installp” :

\$ installp -L -d /export/lpp_source/lpp_source433

dont le résultat sera identique à celui de la commande NIM avec l’option “showres”.

Il est aussi possible de mettre à jour les *filesets* présents uniquement dans le “lpp_source”.

Pour ce la, on peut utiliser un *script* du type :

```
#!/usr/bin/ksh

LPP_NAME=$1
SOURCE_DIR=$2
LPP_TARGET='lsnim -l $LPP_NAME | grep location | awk '{print $3}'

usage () {
    echo «Missing parameter.»
    echo «Usage : lpp_update lpp_source source_dir_update.»
    exit 1
}

check_param () {
    echo «Verification du parametre lpp_source...»
    lsrim $LPP_NAME >/dev/null 2>&1
    if [[ $? -ne 0 ]]
    then
        echo «Error with lpp_source $LPP_NAME paramater...»
        exit 2
    fi
    echo «Verification du source directory de la mise a jour...»
    if [[ -d $SOURCE_DIR ]]
    then
        ls $SOURCE_DIR/.toc >/dev/null 2>&1
        if [[ $? -ne 0 ]]
        then
            echo «$SOURCE_DIR ne contient de filesets ou de .toc»
            exit 3
        fi
    else
        echo «$SOURCE_DIR : repertoire inconnu.»
        exit 4
    fi
}

update () {
    echo «Path vers lpp_source : $LPP_TARGET»
    echo «\nSelections des filesets a mettre a jour...»
    nim -o showres -a installp_flags=L $LPP_NAME | awk -F : '{print $1}' | sort -u > /tmp/list_filesets
    echo «\nListe des filesets disponibles dans la mise a jour...»
    installp -L -d $SOURCE_DIR | awk -F : '{print $1}' | sort -u > /tmp/list_maj
    echo «\nVerification de la disponibilite des filesets et mise a jour...»
    comm -12 /tmp/list_maj /tmp/list_filesets > /tmp/maj_avail
    COUNT='wc -l /tmp/maj_avail'
    echo «$COUNT mise a jour a installer.»
    echo «\nCopie des mise depuis $SOURCE_DIR dans $LPP_NAME...»
    bffcreate -d $SOURCE_DIR -f /tmp/maj_avail -t $LPP_TARGET -vX
    echo «\nVerification du lpp_source $LPP_NAME apres mise a jour...»
    nim -Fo check $LPP_NAME
}
```



```
RESULT=$?
if [[ $RESULT -ne 0 ]]
then
    echo «Verifier votre lpp_source $LPP_NAME.»
else
    exit 0
fi
}
clean_tmp () {
rm /tmp/list_filesets
rm /tmp/list_maj
rm /tmp/maj_avail
}

# Debut du programme principal
# Nettoyage de /tmp en cas d'execution incomplete de lpp_udate

clean_tmp

# Mise à jour.

case $# in
    0 | 1 )
        usage;;
    2 )
        check_param
        update
        clean_tmp;;
    *)
        usage;;
esac
```

Le “spot”

Une fois que le “lpp_source” est à jour, on peut facilement mettre le “spot” au même niveau.

- Mise à jour du “spot” par “smit” :

\$ smitty nim_res_op

- Sélectionner le “spot” que l’on souhaite mettre à jour.
- Sélectionner l’option de mise à jour “update_all”.
- Choisir le “lpp_source” à partir duquel on fait la mise à jour.

On peut valider, une fois terminé.

Vérifier que le résultat est “ok” et qu’aucun message “erreurs” ou “warning” n’est apparu.

- Mise à jour du “spot” en **ligne de commande** :

\$ nim -o cust -a lpp_source=lpp_source433 -a installp_flags=cNg -a fixes=update_all -a auto_expand=no <spot>

Remarque : Il existe bien une option NIM “update”, mais celle-ci s’applique plus particulièrement au “lpp_source” afin d’y ajouter ou d’en retirer des *filesets* particuliers.

Trucs et Astuces

Une fois la mise à jour du “spot” effectuée, vous pouvez vérifier son intégrité grâce à l’opération “check” :

```
$ nim -o check spot433
```

Remarque : L’opération “check” recrée les images de *boot* relatives au spot, ce qui est nécessaire après l’installation d’un nouveau niveau de maintenance. Il est donc conseillé d’effectuer cette opération après chaque mise à jour des ressources.
Dans le cas où cette opération ne recrée pas l’image de *boot* dans “/tftpboot”, il faut configurer le *master* pour recréer automatiquement les images lors de l’opération “check” : **\$ nim -o change -a if_discover=yes master**

Pour connaître le niveau des *filesets* installés dans le spot, passer la commande :

```
$ nim -o showres -a lspp_flags= « l » spot433
```

Fileset	Level	State	Description
Path: /usr/lib/objrepos			
X11.apps.pcmcia	4.3.3.0		COMMITTED AIXwindows PCMCIA GUI Utility
X11.apps.ppm	4.3.3.0		COMMITTED AIXwindows Power Management GUI Utility
bos.64bit	4.3.3.0		COMMITTED Base Operating System 64 bit Runtime
bos.diag.com	4.3.3.0		COMMITTED Common Hardware Diagnostics
bos.diag.rte	4.3.3.0		COMMITTED Hardware Diagnostics
bos.diag.util	4.3.3.0		COMMITTED Hardware Diagnostics Utilities
bos.mp	4.3.3.0		COMMITTED Base Operating System Multiprocessor Runtime
bos.net.nfs.client	4.3.3.0		COMMITTED Network File System Client
bos.net.tcp.client	4.3.3.0		COMMITTED TCP/IP Client Support
bos.net.tcp.smit	4.3.3.0		COMMITTED TCP/IP SMIT Support
bos.powermgt.rte	4.3.3.0		COMMITTED Power Management Runtime

Conclusion

Il est désormais possible de gérer précisément le niveau des ressources et, ainsi, d’avoir toujours :

- un niveau de “lpp_source” et de “spot” en conformité avec les “mksysb”,
- ou ○ un niveau requis pour l’installation des “clients”.

Références Utiles

Pour plus d’information vous pouvez consulter les brochures suivantes :

- **SC23-4113-02** : *Network Installation Guide and Reference*
- **SG24-5524-00** : *Redbook : NIM From A to Z*

ou la documentation en ligne sur le site Internet :

http://publib16.boulder.ibm.com/pseries/en_US/aixins/insgdrf/insgdrf03.htm