

Chapitre 6 Les liens micro-host

6.0 Introduction

Pour accompagner la migration des travaux soumis en Sas Linux vers Sas micro et vice-versa, nous allons exposer et commenter ici les différentes méthodes qui permettent de transférer des fichiers Sas entre ces deux plates-formes.

Toutes les méthodes de transfert que nous allons exposer ont une influence importante sur la charge des réseaux !

Nous allons présenter ici deux méthodes de transfert utilisables à partir de la version Sas Windows, dans un contexte entièrement sous la coupe des versions V9 (en pratique testé entre v9.2.2 windows et 9.1.3 Linux):

1) L'utilisation des procédures Cport et Cimport conjointement au service fourni par FTP, pour écrire ou lire le fichier de transport directement sur Linux (transfert et sauvegarde de fichiers). Cependant comme nous pouvons monter notre ressource personnelle Linux en disponibilité NFS dans Windows et comme Ceda est actif dans le Sas de base sous Windows, il est devenu rare d'effectuer ces transferts !

2) L'utilisation des procédures Upload et Download fournies avec le module Sas/Connect, pour déplacer -sans intermédiaire- les objets Sas d'une plate-forme à l'autre.

6.1 Transfert ordinaire sans Sas Connect: Les procédures CPORT, CIMPORT, utilisées avec FTP en Sas version 9 :

Sens Host vers Micro (réception en 9.2.2 !) :

1ère étape : la création du fichier de transport sur le host

Il est nécessaire d'allouer un fichier séquentiel par une instruction FILENAME pour accueillir les données à transférer. C'est un fichier binaire.

La création à proprement parler du fichier de transport peut se faire en batch ou en interactif, mais on soumettra de toutes les façons un programme similaire (voir pour la syntaxe au chapitre précédent) :

```
libname a "~wielki/temp";  
filename b "~wielki/temp/pays.cport";  
proc cport data=a.pays file=b;  
run ;
```

Il est fortement conseillé de lire la trace sur la LOG de l'exécution demandée. C'est en général à ce stade que se produisent la plupart des erreurs.

Il ne sera cependant pas possible de transférer des programmes compilés (macro ou iml). On recompile toujours les macros quand on change de plateforme ou de version.

C'est aussi pour cette raison qu'il fait toujours garder le programme source, même si les catalogues de macros compilées peuvent maintenant être protégés contre une déconstruction.

2ème étape : la descente du fichier sur le micro dans une session Sas et son importation directe

Sous Sas Windows, il est possible d'utiliser les services de FTP. Dans le cadre de transfert de fichiers Sas entre micro et Linux, FTP va nous permettre d'accéder directement en lecture ou en écriture au fichier de transport distant. Dans cette situation, il n'y a plus de fichier de transport sur le poste micro, d'où un gain d'espace.

Pour effectuer la lecture directe du fichier de transport sur Linux et restauration des objets Sas de départ, sous Sas Windows, il faut avoir recours à la procédure CIMPORT :

```
Libname AV9 V9 "D:\ined\insight\" ;
Filename ftplin ftp "/home/users/wielki/temp/pays.cport"
          host="margaux.ined.fr" user="wielki" rcmd="type i" prompt;
proc cimport data=av9.paysbis infile=ftplin;run;
```

NOTE: 220 Service de transfert de fichiers (FTP)

NOTE: User wielki has connected to FTP server on Host margaux.ined.fr .

WARNING: The transport file is from an earlier SAS release. If this transport file contains non-English data, there may be problems importing national characters. If problems do occur, the transport file will need to be imported again. If the transport file was created by a Unicode SAS session, start a new SAS session, set the ENCODING startup option to UTF8, and redo the import.

Otherwise, set the LOCALE option to match the LOCALE of the SAS session which created the transport file and redo the import.

NOTE: Proc CIMPORT begins to create/update data set AV9.paysbis

NOTE: Data set contains 73 variables and 208 observations.

Logical record length is 584

On remarquera que la version 9.2 de Sas effectue un saut technologique vers l'international car le warning évoque tous les problèmes qui peuvent surgir de par des divergences sur le codepage/encodage du fichier et sur le paramètre LOCAL.

L'instruction FILENAME ... FTP ... permet d'indiquer la localisation et le nom du fichier de transport, les paramètres de connexion sur le serveur (nom d'utilisateur et prompt pour être sollicité pour le mot de passe), et la caractéristique concernant le caractère binaire des données (type i pour image au lieu de a pour ascii).

La connexion au serveur ne s'effectue qu'au moment de l'exécution de la procédure CIMPORT. La déconnexion est réalisée à la fin de la procédure de transfert. Ainsi, l'encombrement sur la machine centrale et sur le réseau sont minimisés.

L'option EXTENDSN=NO du CIMPORT peut être utile en cas d'utilisation du length réduit ($3 \leq l < 8$) sur des variables numériques. La longueur de ces variables numériques raccourcies restera inchangée au lieu de recevoir 1 octet supplémentaire d'extension au titre de sécurité habituelle par Sas dans l'opération de transformation.

Sens Micro (9.2.2) vers Host (9.1.3) :

1ère étape : la création du fichier de transport à partir de Sas Windows et sa réception sur le host

Sous Sas Windows, il y a création du fichier de transport directement sur la machine Linux :

```
libname av9 "d:\ined\insight\";
filename un ftp "/home/users/wielki/temp/pays.cpt" user="wielki"
          host="margaux.ined.fr" prompt rcmd="type i";
proc cport data=av9.pays file=un; run;
```

NOTE: 220 Service de transfert de fichiers (FTP)

NOTE: User wielki has connected to FTP server on Host margaux.ined.fr .

NOTE: Proc CPORT begins to transport data set AV9.PAYS

NOTE: The data set contains 73 variables and 208 observations.

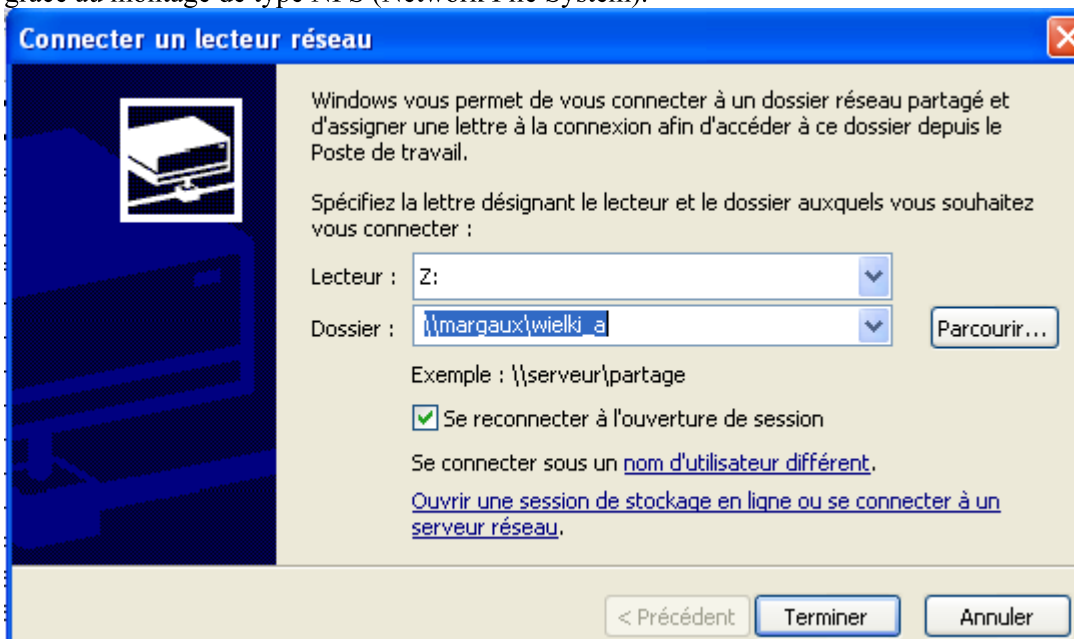
Logical record length is 584.

Il a, bien sûr, fallu remplir l'écran suivant, vu l'option *prompt* du code soumis:



L'instruction FILENAME ... FTP ... permet d'indiquer la localisation du fichier de transport, les paramètres de connexion sur le site central (nom d'utilisateur et mot de passe) et son caractère binaire (TYPE I). La connexion au site central ne s'effectue qu'au moment de l'exécution de la procédure CPORT. La déconnexion est réalisée à la fin de la procédure de transfert. Ainsi, l'encombrement sur le serveur et sur le réseau est minimisé.

Mais dans la pratique, il n'est plus nécessaire de fonctionner ainsi, car il est maintenant possible dans l'Explorateur de monter sur une ressource du poste PC et obtenir une vue du secteur Linux qui vous est réservé grâce au montage de type NFS (Network File System).



Il est ensuite simple de déposer le fichier cport dans cette zone là.

2ème étape : la restauration des objets Sas de départ sous Sas Linux

Cette opération s'effectue à l'aide d'une programme Sas soumis en Sas batch ou en Sas Connect. Il sera nécessaire de soumettre un programme de ce type :

```
filename b "~wielki/temp/pays.cpt";
libname c "~wielki/temp";
proc cimport infile=b data=c.payster;
run;
```

NOTE: Proc CIMPORT begins to create/update data set C.payster

NOTE: Data set contains 73 variables and 208 observations.

Logical record length is 584

NOTE: Data file C.PAYS.DATA is in a format native to another host or the file encoding does not match the session encoding. Cross Environment Data Access will be used, which may require additional CPU resources and reduce performance.

6.2 L'utilisation de Sas Connect (pour ceux qui possèdent Sas windows)

Objectif de Sas/Connect :

Sas/Connect est un module de Sas, qui, à supposer qu'il soit installé en local ainsi que sur le serveur (Windows, Linux, Z-OS). Il permet de réaliser une connexion entre une session Sas micro et des sessions Sas Linux. Celle-ci est techniquement réalisée par l'exécution d'un script qui aboutit à l'ouverture d'une session Sas interactive sur Linux. Cette session distante est alors accessible à partir de l'interface Sas local.

Cette connexion permet ainsi de mettre en œuvre les trois sortes de services suivants:

- le DTS ou Data transfer Service qui donne la possibilité de transférer tous les objets SAS entre les deux sessions micro et Linux, à l'aide des procédures Upload (micro vers Linux) et Download (Linux vers micro)
- la possibilité d'exécuter des programmes Sas sur la machine Linux à partir des données Linux et de recevoir en fin d'exécution l'intégralité de la fenêtre LOG et les sorties éventuelles créées par le programme (OUTPUT, de qualité Linux) [pc=client, linux=serveur de calcul et de données],
- la possibilité de recourir au système de RLS (Remote Library Service) où Sas sur le Pc exécutera les traitements en tenant compte de façon transparente des données sur le serveur

Mise en œuvre de Sas/Connect :

Dans le contexte de l'INED, nous utiliserons le protocole TCP/IP pour réaliser la connexion.

Avant de décrire et illustrer les manières de faire, signalons la possibilité de se connecter à **plusieurs hosts** différents et ce en choisissant un mode de connexion **asynchrone** ou **synchrone**. La différence essentielle est qu'en synchrone vous serez obligé d'attendre la fin de l'exécution avant de pouvoir poursuivre. Ailleurs, certains utiliseront aussi le MPCconnect (multiprocess) .

1) la connexion au serveur statistique : l'exécution du script en situation synchrone

Cette phase permet d'indiquer à Sas, le protocole de communication (TCP), le site host sur lequel on se connecte (LINUX pour Margaux [voir l'autoexec.sas]), ainsi que le script de connexion utilisé (TCPUNIX.SCR). Sur le serveur existe dans ce cas un spawner qui écoute les demandes de connexions et c'est comme cela que les clients peuvent se connecter.

ATTENTION, sur Margaux sont présents les modules Base, Stat, Graph, Ets, Qc, Iml, Insight et Connect . Et aussi Sas Access to pc file format.

La connexion au site host, sur la base des informations précédentes, ne s'effectue qu'au moment de l'exécution de l'instruction SIGNON (ou la rubrique *Signon/Connexion* du menu déroulant Run/Exécuter). L'utilisateur doit alors saisir son login (nom) et son mot de passe associé. L'instruction SIGNON aboutit à l'ouverture d'une session sur la machine Linux choisie et au lancement de Sas en interactif sur Linux.

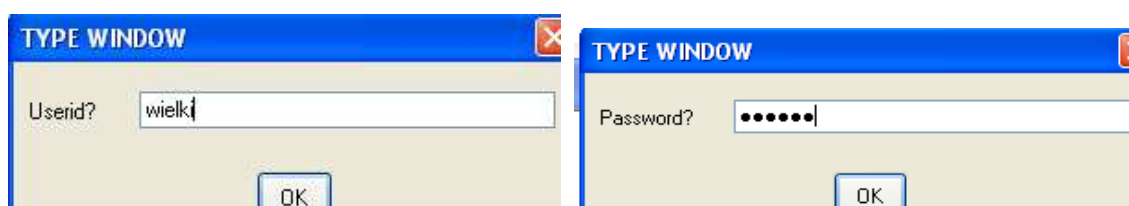
Sous réserve d'avoir pratiqué un AUTOEXEC.SAS de contenu mentionné en page 24, il sera possible en version 9, d'appeler directement cette connexion en indiquant le nom du site choisi dans la rubrique adéquate.

Si vous avez neutralisé l'autoexec.sas par défaut, vous pourrez toujours soumettre le programme suivant après avoir copié (dans d:\My SAS Files\9.2), adapté et personnalisé le script de base TCPUNIX.SCR dans C:\Program Files\SAS\SASFoundation\9.2\connect\saslink\

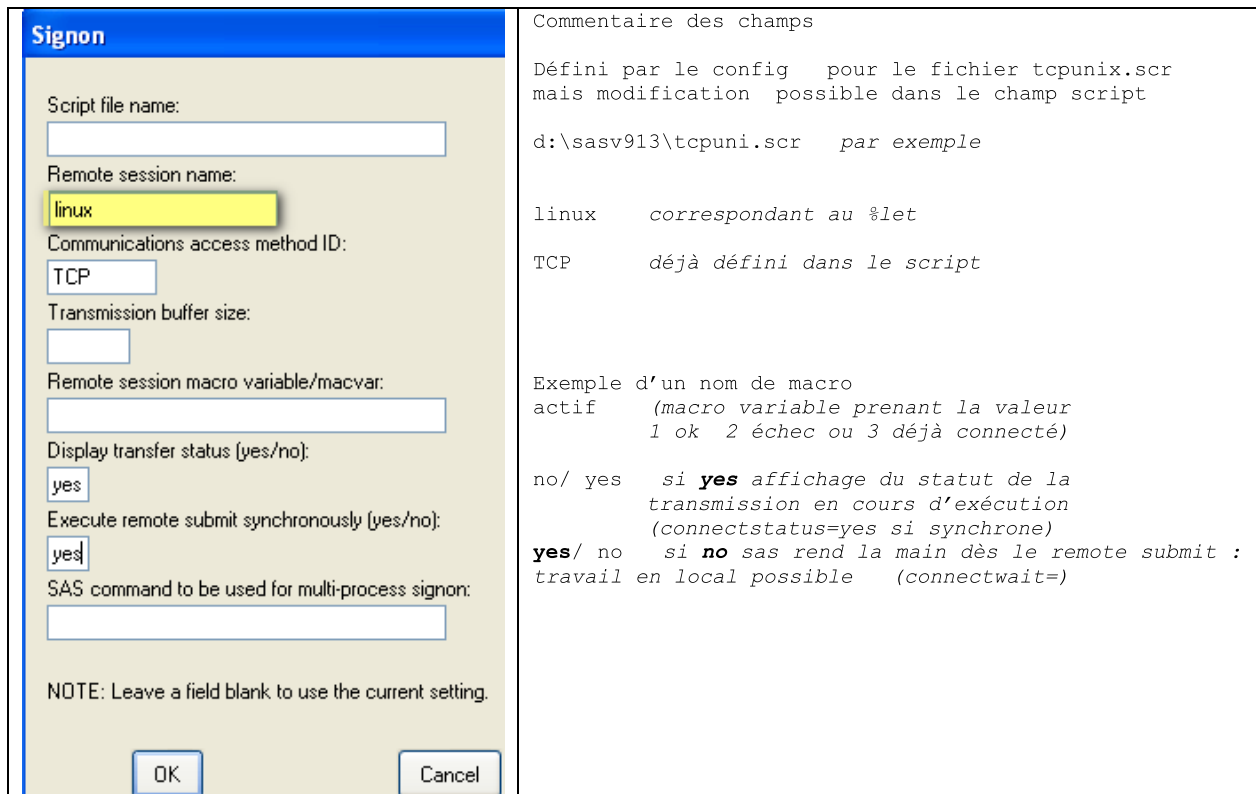
```
%LET LABA=MARGAUX.INED.FR ;
OPTIONS COMAMID=TCP REMOTE=LABA ;
FILENAME RLINK "D:\MY SAS FILES\9.2\TCPUNIX1.SCR" ;
SIGNON LABA;
```

Exemple :

Il suffit de soumettre le SIGNON LINUX ; pour se connecter sur Margaux



et répondre aux deux écrans de sollicitation. Le script se déroule jusqu'au message :
NOTE: Remote signon to LINUX complete.



La connexion synchrone peut aussi s'établir en remplissant le champ *remote session name* de l'écran sollicité par la rubrique *signon* du menu déroulant Run. Seule la rubrique en jaune est nécessaire.

La log confirme bien l'ouverture d'une session v9.1.3 sur Linux

```
NOTE: Remote signon to LINUX commencing (SAS Release 9.02.02MOP011509).
NOTE: Script file 'tcpunix.scr' entered.
NOTE: Logged onto UNIX... Starting remote SAS now.
NOTE: SAS/CONNECT conversation established.
NOTE: Copyright (c) 2002-2003 by SAS Institute Inc., Cary, NC, USA.
NOTE: SAS (r) 9.1 (TS1M3)
      Licensed to UNIV PARIS VI LICENCE RECHERCHE PACK STANDARD, Site xxxxxxxxxxxx.
NOTE: This session is executing on the Linux 2.4.21-63.ELsmp platform.

NOTE: SAS 9.1.3 Service Pack 4
```

```
Bienvenue sur la version 9.1.3 de SAS (sous linux red hat 32 bits)
etc
```

```
NOTE: SAS initialization used:
      real time          0.04 seconds
      cpu time           0.03 seconds
```

```
NOTE: Remote signon to LINUX complete
```

Généralement un signon/connexion peut s'écrire comme instruction :

```
SIGNON REMOTE=nom CONNECTWAIT=yes/no MACVAR=macrovariable
      SCRIPT=fileref USER=nom PASSWORD=xxx (OU PROMPT)
      CONNECTSTATUS=yes/no ;
```

2) une connexion au serveur statistique : l'exécution du script en situation asynchrone

Mais il est possible de se connecter à plusieurs hosts à la fois (Margaux et un éventuel serveur Windows) et de plus continuer à travailler en local pendant que votre Remote submit s'exécute par ailleurs.

Nous devons pour cela remplir plus de rubriques significatives de l'écran: au minimum les rubriques WAIT= et STATUS= outre l'obligatoire REMOTE=.

Exemple :

<p>Script file name: <input type="text"/></p> <p>Remote session name: <input type="text" value="linux"/></p> <p>Communications access method ID: <input type="text"/></p> <p>Transmission buffer size: <input type="text"/></p> <p>Remote session macro variable/macvar: <input type="text"/></p> <p>Display transfer status (yes/no): <input type="text" value="yes"/></p> <p>Execute remote submit synchronously (yes/no): <input type="text" value="no"/></p> <p>SAS command to be used for multi-process signon: <input type="text"/></p>	<p>J'ouvre une session asynchrone de nom Linux (donc sur Margaux) en utilisant le script de connexion tcpunix.scr par défaut puisque je n'ai rien indiqué.</p> <p>Cette session sera asynchrone parce que j'ai répondu no à la question sur le mode, mais j'aurai aussi, vu ma réponse yes à la question sur le display, la possibilité de voir une fenêtre de transfert sur la progressivité de mes transferts issus de download et upload, enfin, je pourrai suivre le déroulement d'une session host sur une ou deux fenêtres distinctes (display log et display output) avant de rapatrier en local toutes ses informations qui iront s'intégrer dans le flot local (après la fin de la procédure en cours) au moment où je le déciderai.</p>
---	---

3) une connexion automatique au serveur statistique avec soumission

```
options autosignon;  
rsubmit linux persist=no;  
proc print data=sashelp.class;run;  
endrsubmit linux;
```

Ici nous encadrons le code soumis à Linux sans avoir fait de signon manuel. Du fait de l'option autosignon, la sollicitation du nom etc. se fait de manière automatique et le persist=no enclenche la déconnexion automatique.

4) le remote submit dans la situation d'une session synchrone/asynchrone

Normalement dans une session **synchrone ouverte**, la soumission du code sélectionné destiné au host à distance s'encadre par l'action de la rubrique *Remote submit* du menu déroulant Run (et spécification de la *Remote Session Name* sur la fenêtre de dialogue ouverte : en l'occurrence *linux* dans notre cas simple.

Ou par l'encadrement du code à soumettre dans la fenêtre programme par :

```
RSUBMIT ;  
CODE... ;... ;... ;RUN ;  
ENDRSUBMIT ;
```

soumis en Submit simple

Vous devez attendre que votre code s'exécute avant de pouvoir actionner votre Sas local.

Dans le cadre d'une session **asynchrone**, de nouvelles possibilités se présentent dues au fait qu'on puisse avoir plusieurs programmes qui se déroulent en même temps sur des host différents (Margaux et un autre serveur dans le cadre de l'Ined). De plus, vous avez la main pour continuer à travailler en local sous Sas.

Remote Submit	
Remote session name: linux	
Remote session macro variable name: <input type="text"/>	
Display transfer status (yes/no): <input type="checkbox"/>	
Execute remote submit synchronously (yes/no): <input type="checkbox"/>	No pour contrecarrer le mode par défaut yes ou synchrone

La session s'ouvre avec un message :
 NOTE: Remote submit to LINUX complete.
 NOTE: Background remote submit to LINUX in progress.
 NOTE: Remote submit to LINUX commencing.

Si j'actionne la rubrique *Remote Display* du menu déroulant Run, j'obtiens une nouvelle fenêtre informative sur le déroulement de mon programme sur le host désigné :

```

S RDISPLOG: LINUX
NOTE: Remote submit to LINUX commencing.
MPRINT(RSUBMIT):  proc contents data=work._all_;
MPRINT(RSUBMIT):  run;
3  proc contents data=work._all_;run;

WARNING: No matching members in directory.
NOTE: PROCEDURE CONTENTS used (Total process time):
      real time          0.03 seconds
  
```

RDISPLOG: LINUX ... S RDISPLOG: LINUX ...

accompagnée d'une seconde fenêtre Rdispout le cas échéant s'il y a déjà des résultats du type Output.

```

S RDISPLOG: LINUX
FORM FEED>-----
                                The SAS System      16:53 Friday, January 29, 2010  3
                                The CONTENTS Procedure
                                Directory
                                Libref          WORK
  
```

RDISPLOG: LINUX ... S RDISPLOG: LINUX ...

L'instruction `RDISPLOG LINUX ;` donne le même résultat.

A la fin de l'exécution de tout votre programme host, vous pourrez rapatrier dans les fenêtres respectives, l'ensemble des résultats et messages par l'action de la rubrique *Remote Get* du menu déroulant Run ou la soumission de `RGET LINUX ;`

Remote Display	Remote Get
Remote session name: linux	Remote session name: linux
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	<input type="button" value="OK"/> <input type="button" value="Cancel"/>

Cette action ne peut s'exécuter qu'une seule fois, car, dès l'usage du Remote get, vous vous retrouvez dans la situation d'une session synchrone.

Il existe la possibilité de nommer une macro-variable liée à l'état du Rsubmit : celle-ci ne retournera rien s'il y a eu une erreur de syntaxe dans le Rsubmit, 0 si le Rsubmit est fini, 1 s'il ne s'est pas exécuté et 2 s'il est toujours en cours. Vous pouvez donc en la testant éviter de 'casser' le caractère asynchrone de votre session qui vous laisse les mains libres en Sas local.

Syntaxe : RSUBMIT <REMOTE SESSION ID> CONNECTWAIT=YES|NO MACVAR=XXXX
CONNECTSTATUS=YES|NO SYSRPUTSYNC=YES|NO; ENDRSUBMIT ;

SYSRPUTSYNC=YES forcera l'exécution du %SYSRPUT dès soumission dans le cadre d'une situation asynchrone. Pour les autres aspects liés au macro langage, voir mon autre brochure.

6.3 Fermeture de la session Sas Connect :

La fermeture de la session Sas sur Linux et du lien entre le micro et le serveur est réalisée par l'instruction SIGNOFF (ou la rubrique *Signoff* du menu Run). Le lien reste effectif jusqu'à l'exécution du SIGNOFF.

Il est très important de toujours veiller à se déconnecter "proprement" par un SIGNOFF pour ne pas encombrer les réseaux et la machine Linux. Un oubli laissera des traces dans la saswork Linux.

Pour effectuer ce signon, vous passerez l'instruction

SIGNOFF ; OU SIGNOFF LINUX ;

6.4 DTS ou Data Transfer Service par Sas Connect : Les procédures UPLOAD et DOWNLOAD

Attention : la soumission des instructions sur le serveur Linux se fait par un *remote submit* et non par un *submit* ! Erreur fréquente !

Les LIBNAMES doivent être exécutés avant. Il y a en général un libname à définir en local (en *submit*) et un autre à définir sur le host (en *Rsubmit*)

a) La procédure Download

Syntaxe simplifiée valable pour le download ou le upload :

```
PROC DOWNLOAD/UPLOAD <AFTER= date CONNECTSTATUS=NO>
  DATA= Sasdataset      OUT= Sasdataset      /* pour une table */
  INLIB= libref          OUTLIB= libref        /* pour une librairie */
  INCAT= libref.catalog  OUTCAT= libref.catalog /* pour un catalogue */
  INFILE= file-specifier OUTFILE= file-specifier /* pour un fichier non-sas */
  <options> ;
  instructions éventuelles de contrôle complémentaires
  SELECT member-list </MEMTYPE= mtype>;
  EXCLUDE member-list </MEMTYPE= mtype>;
  WHERE .... ; *uniquement dans la cas d'UNE table sas ;
Run ;
```

Elle réalise les transferts dans le sens Margaux vers micro.

Les <options> de la procédure pour les datasets sont e.a:

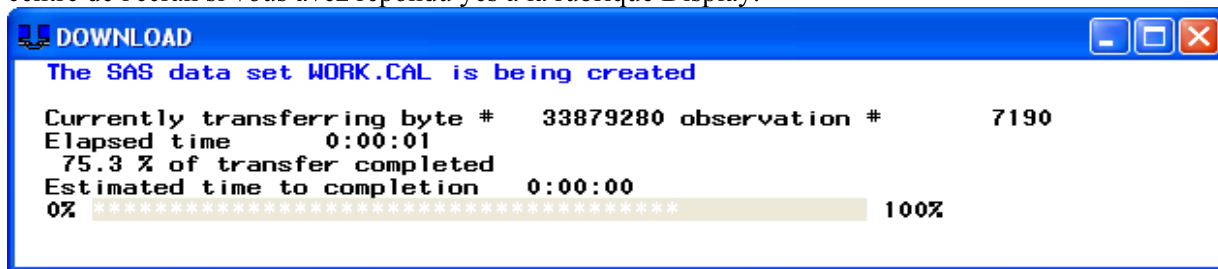
- EXTENDSN=NO pour éviter l'ajout d'un octet au transfert à chaque numérique court (<8 octets).
- INDEX=NO empêchera la recréation de l'index existant

Les <options> de la procédure pour les librairies sont les mêmes que pour les tables avec en plus MEMTYPE=(mtype-list). Les <options> pour les catalogues sont EXTENDSN=NO et ENTRYTYPE=.

Et celle pour les fichiers non-sas pour empêcher une traduction ascii-ansi- ou non testé UTF8 : BINARY !

Exemple de transfert :

Pendant le transfert, une fenêtre, indiquant la durée écoulée et le pourcentage du transfert réalisé, apparaît au centre de l'écran si vous avez répondu yes à la rubrique Display.



NOTE: Remote submit to LINUX commencing.

```
5 libname a "~wielki/OLD";
```

NOTE: Libref A was successfully assigned as follows:

```
Engine: V9
Physical Name: /home/users/wielki/OLD
```

```
6 proc download data=a.calendrier out=work.cal;run;
```

NOTE: Data file A.CALENDRIER.DATA is in a format native to another host or the file encoding does not match the session encoding. Cross Environment Data Access will be used, which may require additional CPU resources and reduce performance.

NOTE: Download in progress from data=A.CALENDRIER to out=WORK.CAL

NOTE: 44985464 bytes were transferred at 25258544 bytes/second.

NOTE: The data set WORK.CAL has 9547 observations and 621 variables.

NOTE: Downloaded 9547 observations of 621 variables.

NOTE: The data set A.CALENDRIER has 9547 observations and 621 variables.

NOTE: PROCEDURE DOWNLOAD used (Total process time):

```
real time      2.03 seconds
cpu time       0.73 seconds
```

NOTE: Remote submit to LINUX complete.

b) La procédure Upload

Elle réalise les transferts dans le sens micro vers Linux. La syntaxe est identique à la procédure Download au changement de nom de procédure près. Dans le sens micro vers Linux, il est préférable de vérifier avant transfert que l'espace disponible est suffisant. La procédure Upload ne gère pas ces problèmes. Ici le retour vers Linux de la table calendrier.

NOTE: Remote submit to LINUX commencing.

```
8 proc upload data=work.cal out=a.retourcal;run;
```

NOTE: Upload in progress from data=WORK.CAL to out=A.RETOURCAL

NOTE: 44985464 bytes were transferred at 44277030 bytes/second.

NOTE: The data set WORK.CAL has 9547 observations and 621 variables.

NOTE: Uploaded 9547 observations of 621 variables.

NOTE: The data set A.RETOURCAL has 9547 observations and 621 variables.

NOTE: PROCEDURE UPLOAD used (Total process time):

```
real time      1.06 seconds
cpu time       0.65 seconds
```

NOTE: Remote submit to LINUX complete.

Les transferts de tables indexées

Sauf usage de l'option INDEX=NO, par défaut, les index et les tables sont transférés ensemble par les procédures UPLOAD et DOWNLOAD.

Une exception toutefois, l'option DATA= et OUT= pour le transfert *d'une seule table*, ne conserve pas les index.

Il faut utiliser dans ce cas la syntaxe de transfert d'une librairie avec l'instruction complémentaire SELECT pour sélectionner la table.

6.5 Utilisation de Sas/Connect pour exécuter certains traitements interactifs sur le Host LINUX

Une fois la session établie, on peut exécuter sur le Host Linux tous les traitements Sas classiques sur des données Sas Linux. Les programmes sont saisis dans la fenêtre de l'Editor de Sas Windows et envoyés au site central par la commande **R**SUBMIT du menu déroulant Run. L'exécution s'effectue intégralement sur LINUX, mais le contenu des fenêtres LOG et OUTPUT est visualisable sur micro dans la LOG et l'OUTPUT de Sas micro.

Dans ce mode d'utilisation, Sas Windows est utilisé comme unterface frontale du serveur Sas Linux. Les traitements sont saisis sur micro et envoyés pour exécution sur le site central. Les sorties LOG et OUTPUT sont visualisables dans la session Sas micro. Ainsi, les listings peuvent être sauvegardés sur micro ou imprimés sur une imprimante classique locale.

Les traitements sont effectués en interactif et les sorties sont de qualité Linux, puisque c'est la session Linux qui exécute les traitements. Les options Sas valides sous Linux agissent au lieu de celles de Sas Windows. Pensez donc à passer les options adéquates pour vos futures sorties. Sachez aussi qu'actuellement les minuscules accentuées sont aussi autorisées en Linux comme modalité de variable caractère ou entre quotes en tant que texte (par ex dans un proc format). Il n'y a pas de nécessité de table de transcodage tant que vous n'êtes pas en UTF8.

Pas d'appel autorisé à Viewtable. Mais le montage (en nfs Network File System de RedHat) vous permet de voir vos fichiers si monté dans l'explorateur de Windows et que par conséquent l'usage du Sas Universal Viewer vous dépannera.

En revanche, toutes les étapes DATA, les procédures **FREQ**, **CONTENTS**, **PRINT**, **TABULATE**, **MEANS**, **REG**, ... et toutes les opérations qui manipulent des tables Sas ou qui produisent des résultats agrégés sont parfaitement utilisables. Les procédures de Sas Graph pourront afficher un graphique imparfait par l'intermédiaire du device **GRLINK**. Leur impression correcte n'est pas garantie. Je n'ai pas encore pu tester les nouvelles procédures graphiques **SG** et autres nouveautés dans ce cadre de soumission à distance, car le Linux **INED** possède encore une version 9.1.3 sp4 ce jour.

Exemple : utilisation de la procédure TABULATE.

Le programme est exécuté par la commande **R**SUBMIT ou la rubrique *Remote submit* du menu Run/Exécuter. Le résultat, de qualité Linux, est retourné dans la fenêtre **OUTPUT** de Sas Windows sous les réglages suivants préalables : la fonte de la fenêtre Output a été définie en « sas monospace ». La fenêtre **LOG** du micro contient le texte suivant et la fenêtre locale Output a récupéré un tableau avec bordures.

```
options autosignon;
rsubmit linux persist=no;
proc tabulate data=sashelp.class formchar='8082838485868788A689BC'x;
class sex ;var age;
table sex all, age*mean;
run;
endrsubmit ;
```

Les meilleures bordures ont été obtenue en activant l'option `FORMCHAR = "8082838485868788A689BC"X` Mais n'oubliez pas que ceci n'est que de la visualisation sur votre écran Windows à l'intérieur de Sas Windows de quelque chose qui s'est passé sur Linux.

Dans la Partie suivante, vous découvrirez qu'on peut s'y prendre autrement et voici un exemple qui montre la quintessence d'un résultat dans Excel mais créé sur Linux !

```
options autosignon;
rsubmit linux persist=yes;
ods listing close;
ods noresults;
ods html file="~wielki/temp/connectexcel.htm";
```

```

title " ";
proc tabulate data=sashelp.prdsale;
class product country;
classlev product /
style= {htmlstyle=' mso-rotate:40 ' cellheight=1.5 cm };
var actual predict;
table country="" * (actual predict) * F=6.1, product="" * mean="";
run;
ods html close;
ods results;
ods listing;
endrsubmit ;

```

donne

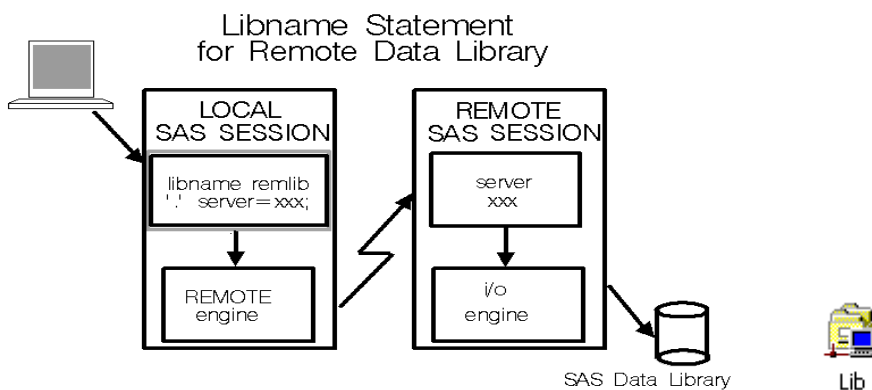
		BED	CHAIR	DESK	SOFA	TABLE
CANADA	Actual Sales	497.2	523.3	543.6	522.2	486.5
	Predicted Sales	460.6	487.5	514.5	476.3	488.4
GERMANY	Actual Sales	480.6	490.7	505.2	573.5	512.5
	Predicted Sales	456.2	459.1	465.0	515.8	516.0
U.S.A.	Actual Sales	501.8	530.6	505.7	452.0	482.3
	Predicted Sales	519.3	471.3	543.4	470.9	513.0

à condition de bien l'ouvrir dans Excel2007 par le biais de l'ouverture comme source Web. L'ODS est vu au chapitre 9.

6.6 Utilisation de Sas/Connect dans le cadre du service Remote Library :

Un autre service, peu connu, permet d'accéder à des ressources Sas situées sur le remote host et de travailler cependant en local. L'intérêt reste limité à l'utilisation de modules installés sur le poste Windows uniquement et sur des données distantes.

Dans cette situation, ce sont les données qui bougent fréquemment en transit, mais elles restent stockées sur le host : ce qui peut engendrer un assez gros trafic réseau. Ces données peuvent être des tables Sas, des vues Sas, des vues SQL ou des access views. Cependant le gros avantage est qu'il n'y a pas multiplication des données qui restent localisées à un seul endroit.



Vous préciserez le nom du serveur dans un libname spécial et Sas cherchera sa correspondance avec le nom d'une liaison Sas Connect déjà établie.

```
LIBNAME LIB REMOTE "~wielki/OLD" SERVER=LINUX ;
```

est un exemple et permet de passer ensuite **toujours en submit** un pgm.

```
NOTE: Remote submit to LINUX complete.
7 LIBNAME LIB REMOTE "-wielki/OLD" SERVER=LINUX ;
NOTE: Libref LIB was successfully assigned as follows:
Engine: REMOTE
Physical Name: /home/users/wielki/OLD
```

Il est bien sûr possible de travailler sur des données de quelqu'un d'autre (autre répertoire de base) : Ecriture générale

```
LIBNAME libref REMOTE <"sas-data-library"> SERVER=remote-session-id <ACCESS=READONLY>
<OUTREP=format> <SLIBREF=server-libref> <engine-options> ;
```

L'option OUTREP= permet d'obliger la création de nouveaux fichiers sous une architecture autre que celle du host. Mettre OUTREP=LINUX est redondant, car par défaut le remote host crée déjà naturellement cette nouvelle table sous son format natif.

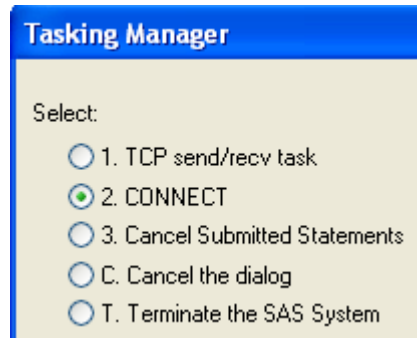
SLIBREF=*nom* permet de spécifier le libref qui sera utilisé par la session host pour identifier une librairie Sas dont le nom physique n'est pas défini et dont le nom local de libref sera différent. Typiquement, nous pourrions ainsi identifier la vraie work *locale* de la work *du host* dans un contexte où ce qui est soumis l'est uniquement en **submit** :

```
LIBNAME RWORK REMOTE SLIBREF=WORK SERVER=LINUX ;
```

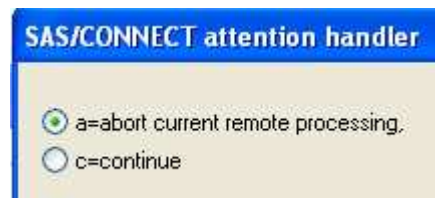
Les <engine options> sont rares et ardues. ROPTIONS= " PS=75 LS=100 " passera des options. L'utilisation de catalogue (par exemple de formats) sera rendu difficile de par leur représentation différente en Windows et Linux.

6.7 Interrompre Sas Connect :

Vous pouvez éprouver le besoin d'arrêter un processus Sas en cours sur le host. Faire Ctrl + Pause et cochez une rubrique de la fenêtre dialogue.



Faire le choix 3, mais il est aussi possible de faire le choix 2 et confirmer alors un abort du Sas connect



Au bout de quelques instants, la log indiquera un signoff automatique. Si cela ne marche pas faire le 1 qui forcera une fin plus brutale.