

SAS ET SON ENVIRONNEMENT

L'environnement SAS est riche et complexe, se basant sur l'environnement système où il est installé d'une part puis sur les configurations SAS d'autre part ; il n'est pas toujours évident de s'y retrouver.

Cet article dans un premier temps théorique, ne prétend pas à l'exhaustivité mais vous permettra de mieux appréhender les liens entre SAS et son environnement.

Dans un second temps vous seront proposés quelques exemples.



Caractéristiques :

Catégories : Base
OS : Windows, Unix, z/OS
Version : SAS® 9.4
Vérifié en septembre 2013

Sommaire

1.	Les Racines Systèmes : L'Environnement SAS	2
1.1.	Les variables d'environnement	2
1.2.	Sassw.config	3
1.3.	Sasenv, level_env, appservercontext_env	4
1.4.	Foundation et clients vs Plateforme	6
2.	L'Arbre: La Configuration SAS	7
2.1.	Le tronc : sasv9*.cfg	8
2.1.1.	Kesako : sasv9*.cfg	8
2.1.2.	A plusieurs niveaux	9
2.1.	L'automatisation : Autoexec.sas	10
2.1.1.	Présentation	10
2.1.2.	Possibilités	10
2.1.3.	Pièges	11
2.2.	Bibliothèque Sasuser	11
2.2.1.	Présentation	11
2.2.2.	Possibilités	12
2.3.	SAS Options	12
2.3.1.	Présentations	12
2.3.2.	Sauvegarde et restauration d'une liste d'options	13
2.3.3.	Restriction d'options	13
3.	Etudes de cas :	14
3.1.	Montée de version du JRE	14
3.2.	Montée de version d'un client SGBD et plus précisément Oracle	15
3.3.	Changement de localisation de la WORK	17
3.4.	Changement des paramètres email	18
3.5.	Définition d'un autoexec	19
4.	En cas de problème	20
5.	Liens utiles	20
6.	Conclusion	21

1. LES RACINES SYSTÈMES : L'ENVIRONNEMENT SAS

L'environnement système SAS, tel des racines, est la fondation dont ce dernier a besoin pour fonctionner.

1.1. Les variables d'environnement

Nous appellerons ici environnement système la couche que l'on pourrait représenter entre le système d'exploitation et le logiciel.

Le fonctionnement de ce dernier passe avant tout par des variables d'environnement, ces dernières peuvent être définies à un niveau général de la machine ou à un niveau utilisateur.

Par la définition de ces variable dynamiques, il est alors possible de garder en mémoire la disposition du clavier, le nom réseau de la machine, les chemins où chercher un exécutable, la localisation du dossier « home » de l'utilisateur courant, etc.

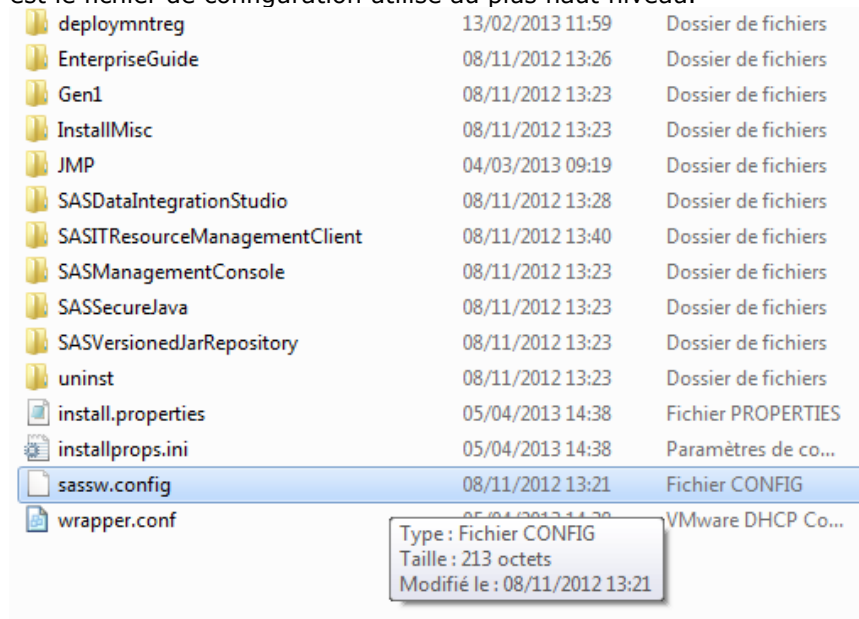
Le fonctionnement est similaire sous un système de type Windows ou de type Unix, il s'agit dans un premier temps de définir la variable puis dans un second temps de l'utiliser.

Ci-après un tableau rappelant les opérations de base sur les variables d'environnement :

<u>Système d'exploitation</u>	<u>Unix</u>	<u>Windows</u>
<i>Afficher toutes les variables définies</i>	env	set
<i>Afficher le contenu d'une variable</i>	Echo \$NomVariable	set NomVariable ou echo %NomVariable%
<i>Définir une variable</i>	NomVariable=valeur Export NomVariable	set NomVariable=valeur
<i>Définir une variable en y incluant sa valeur précédente</i>	NomVariable=\$NomVariable:valeur Export NomVariable	NomVariable=%NomVariable%;valeur
<i>Supprimer une variable</i>	NomVariable=	set NomVariable=

1.2. Sassw.config

Le fichier sassw.config se trouve à la racine de SAS_HOME (répertoire racine d'installation SAS) et est le fichier de configuration utilisé au plus haut niveau.



Ce dernier contient notamment ces variables :

- JRE_HOME contient le chemin du JRE (Java Runtime Environment) utilisé par les clients java SAS et est défini lors de l'installation.
- SAS_HOME contient le chemin du répertoire où ont été installés les logiciels SAS, il est défini lors de l'installation.
- DPLMTREGLOC contient le chemin du Registre de déploiement (Deployment Registry).

En effet toute installation de produit ou de correctif passe par ce registre de déploiement afin d'enregistrer toute installation relative à SAS.

- VJRHOME correspond au chemin où se trouve SAS Versioned Jar Repository (VJR) qui est le répertoire où tous les .jar utilisés par SAS sont stockés.
Pour rappel les fichiers .jar sont des « archives java ». Ce format permet de grouper des définitions de classes et des métadonnées constituant l'ensemble d'un programme.

Il est à noter que le fichier sassw.config est appelé par les clients SAS basés sur SAS Application Launcher comme Data Integration Studio, SAS Management Console, SAS Enterprise Miner ou encore SAS Deployment Manager.

Un changement d'emplacement du JRE, suite à une mise à jour par exemple sur un poste client, doit par conséquent être répercuté dans ce fichier si vous souhaitez que cette modification soit prise en compte.

Java étant multi plateformes le fonctionnement est similaire sous Windows ou sous un système Unix.

1.3. Sasenv, level_env, appservercontext_env

Ces scripts, portant l'extension « .bat » sous Windows ou « .sh » sur les systèmes Unix, sont appelés et exécutés à chaque lancement de SAS Foundation ou d'un serveur logique SAS (serveur de métadonnées, Object Spawner, etc.).

En effet, « sasenv » est utilisé par l'environnement SAS Foundation.

Sous Unix il est alimenté lors de l'installation et contient les chemins nécessaires aux bibliothèques par exemple. Si une modification est nécessaire au bon fonctionnement de SAS, elle doit être faite dans sasenv_local.

Exemple de fichier sasenv_local dans un environnement Unix :

```
JAVA_HOME=""

if [ "$SAS_JAVA_HOME" != "" ] ; then
    if [ -f ${SAS_JAVA_HOME}/bin/java -a -d ${SAS_JAVA_HOME}/lib/ppc/j9vm ] ; then
        JAVA_HOME=${SAS_JAVA_HOME}
    fi
fi

if [ "$JAVA_HOME" = "" ] ; then
    JAVA_HOME=/sas/applis/ITRM33/install/jre1.6_SR8_FP1+IZ77436
fi

export JAVA_HOME

if [ "$LIBPATH" != "" ] ; then
    LIBPATH=$JAVA_HOME/lib/ppc:$JAVA_HOME/lib/ppc/j9vm:${SAS_INSTALL_ROOT}/sasexe:$LIBPATH
else
    LIBPATH=$JAVA_HOME/lib/ppc:$JAVA_HOME/lib/ppc/j9vm:${SAS_INSTALL_ROOT}/sasexe
fi

export LIBPATH

AIXTHREAD_COND_DEBUG=OFF
export AIXTHREAD_COND_DEBUG

AIXTHREAD_MNRRATIO=1:1
export AIXTHREAD_MNRRATIO

AIXTHREAD_MUTEX_DEBUG=OFF
export AIXTHREAD_MUTEX_DEBUG

AIXTHREAD_RWLOCK_DEBUG=OFF
export AIXTHREAD_RWLOCK_DEBUG

AIXTHREAD_SCOPE=S
export AIXTHREAD_SCOPE

LDR_CNTRL=LOADPUBLIC
export LDR_CNTRL

if [ -f /sas/applis/ITRM33/install/Secure/bin/secureenv ] ; then
    . /sas/applis/ITRM33/install/Secure/bin/secureenv
fi

if [ -f ${SAS_INSTALL_ROOT}/bin/sasenv_local ] ; then
    . ${SAS_INSTALL_ROOT}/bin/sasenv_local
fi
```

Sous Windows, SAS Foundation n'a pas besoin de ce fichier car SAS Foundation se base sur les variables d'environnement définies à l'installation.

Level_env et appservercontext_env sont utilisés par la plateforme SAS sous Windows et sous Unix. Ils sont utilisés au lancement du serveur de métadonnées, de l'Object Spawner, du serveur OLAP, d'un serveur Workspace ainsi que tous les autres serveurs SAS.

Il est à noter que le nombre, l'emplacement ou même l'existence des serveurs SAS dépend de la configuration mise en place et de la licence associée.

Exemple de fichier level_env.sh dans un environnement Unix :

```
#!/bin/sh
#
# level_env.sh
#
# Set environment variables relative to the configuration level structure
#
LEVEL_ROOT=/sas/applis/ITRM33/config/Lev2
UTILITIES=/sas/applis/ITRM33/install/SASDeploymentManager/9.3/products/cfgwizard_93535_prt_xx_sp0_1/Utilities
DEPLOYWIZ=/sas/applis/ITRM33/install/SASDeploymentManager/9.3/products/deploywiz_93530_prt_xx_sp0_1/deploywiz
SAS_HOME=/sas/applis/ITRM33/install
JAVA_JRE_COMMAND=/sas/applis/ITRM33/install/jre1.6_SR8_FP1+I277436/bin/java
SASVJR_HOME=/sas/applis/ITRM33/install/SASVersionedJarRepository
SASVJR_REPOSITORYPATH=/sas/applis/ITRM33/install/SASVersionedJarRepository/eclipse
SASWebInfrastructurePlatform_HOME=/sas/applis/ITRM33/install/SASWebInfrastructurePlatform/9.3
SASROOT=/sas/applis/ITRM33/install/SASFoundation/9.3
SAS_COMMAND=$SASROOT/sas
JDK_HOME=/usr/java6_64
JBOSS_HOME=/sas/applis/ITRM33/jboss/jboss-4.2.3.GA/
```

Ces fichiers contiennent entre autres : le chemin de la configuration (LEVEL_ROOT), le chemin de l'installation de SAS (SAS_HOME), la commande pour lancer une JRE (JAVA_JRE_COMMAND), le chemin de SAS VJR évoqué précédemment, le chemin où se trouve SAS (SAS_ROOT), le chemin du JDK (JDK_HOME) et le chemin du serveur web paramétré (JBOSS_HOME, WLS_HOME, WAS_HOME).

Le but en référençant ces scripts est de permettre à un environnement SAS d'avoir une configuration cohérente.

En effet, si un problème d'environnement général à la plateforme venait à survenir sous Windows par exemple, pour le résoudre :

- il suffirait de modifier les fichiers level_env.bat ou level_env_usermods.bat, ces derniers se trouvant à la racine du dossier LevX (où X est compris entre 1 et 9) ou si le problème n'a lieu que pour un contexte en particulier,
- la modification concernerait les fichiers appservercontext_env.bat ou appservercontext_env_usermods.bat se trouvant à la racine de chaque contexte.

SASMeta et SASApp sont des exemples de contextes.

Pour rappel, concernant les scripts utilisés dans le cadre d'une configuration, si une modification est nécessaire ou souhaitée, elle doit être faite dans le fichier suffixé par « _usermods », ceci afin de ne pas perdre les modifications lors d'une mise à jour de SAS par exemple.

En effet, une mise à jour peut avoir à remplacer les fichiers fournis à l'installation.

Exemple: D:\SAS\Config\Lev1\level_env_usermods.cfg

1.4. Foundation et clients vs Plateforme

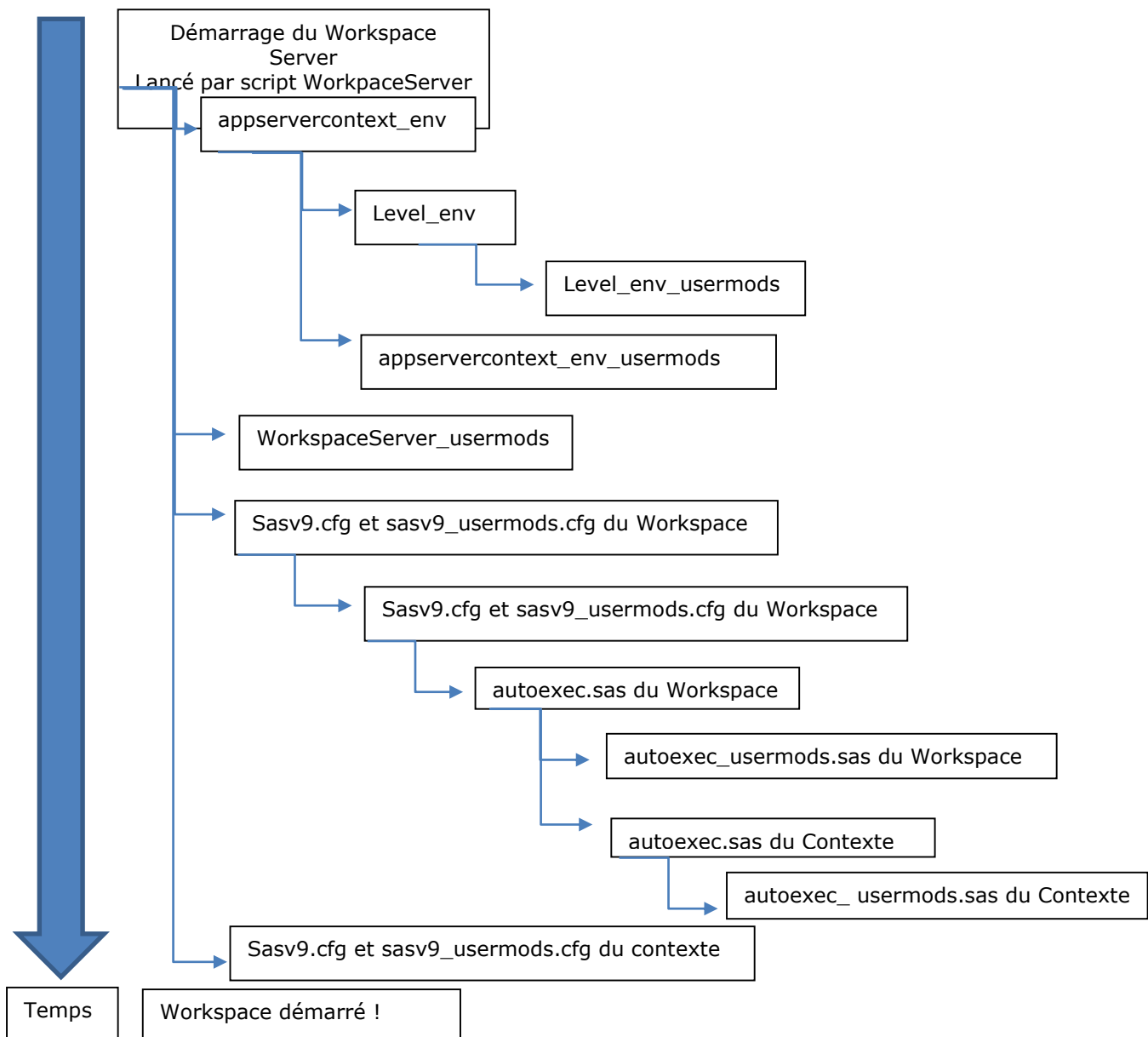
Comme nous avons pu le voir ci-dessus, une configuration de type Clients Java ou SAS Foundation ne s'articule pas exactement de la même façon qu'une configuration de la plateforme SAS.

Tandis que la configuration des clients se base essentiellement sur le fichier `sassw.config` (ainsi que le `.ini` dans le dossier du client) et celle de SAS Foundation sur les différentes variables définies à l'installation ou le fichier `sasenv`, la configuration plateforme, sous Windows ou sous Unix, fait appel, à chaque action, aux variables d'environnement définies dans les fichiers `level_env` et `appservercontext_env`.

Une plateforme n'ayant pas d'interface utilisateur directe, il s'agit d'éviter par exemple un cas de modification d'une variable par un administrateur système, qui pourrait passer inaperçue mais avoir des conséquences graves sur une plateforme.

Il a donc été mis en place un système de scripts appelés en cascade, ces derniers facilitent la gestion de la plateforme (démarrage, arrêt, statut, etc.) sans avoir à se préoccuper de savoir si l'environnement est bien défini, étant donné que chacun des scripts présents fait appel aux différents fichiers nécessaires à son fonctionnement (`level_env` et `appservercontext_env`).

Ci-après une illustration pour le lancement d'une session Workspace :



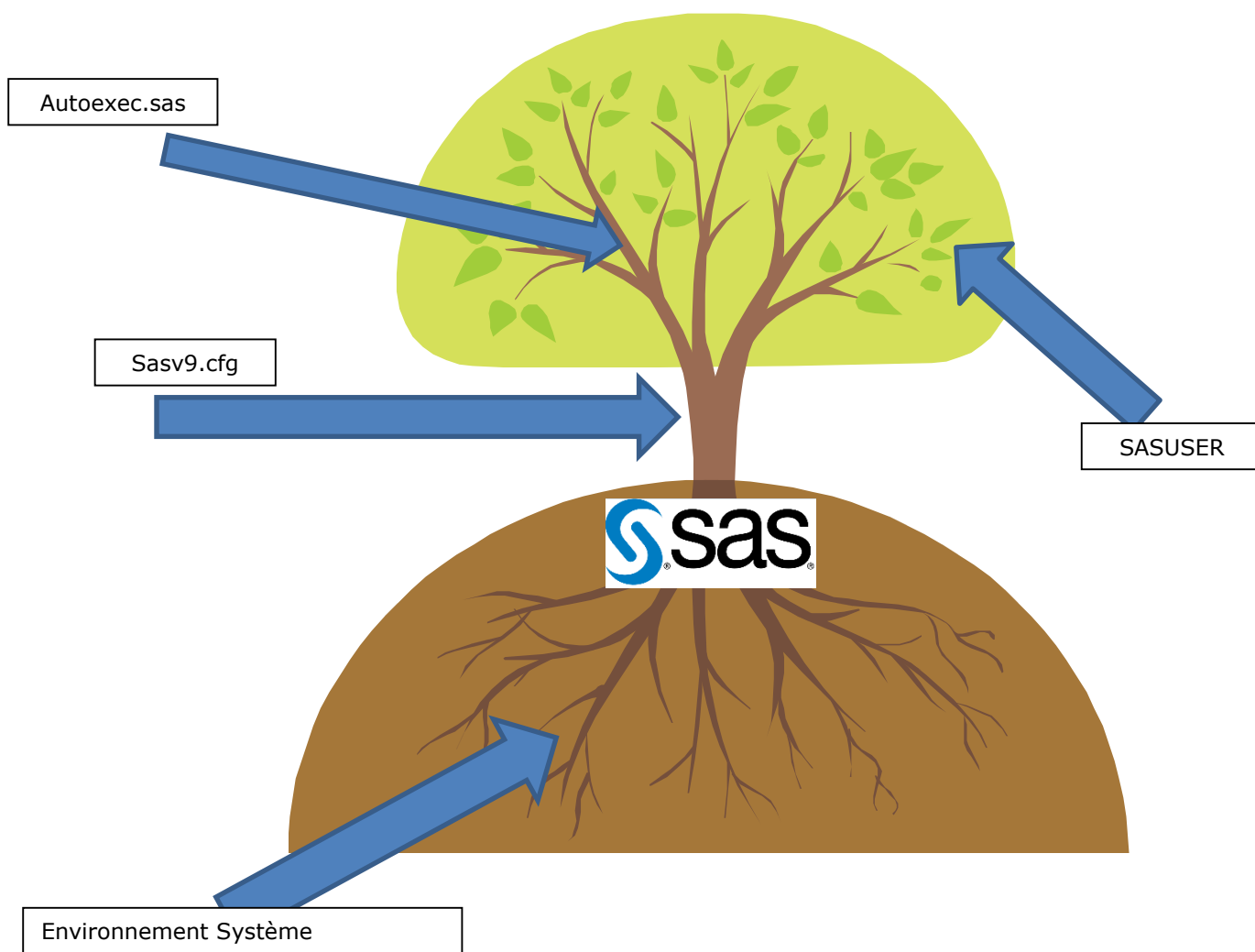
2. L'ARBRE: LA CONFIGURATION SAS

Au lancement de SAS Foundation ou de la plateforme SAS, une fois la partie système chargée, intervient alors le chargement des fichiers de configuration (2.1), du profil utilisateur (2.2), d'éventuels fichiers autoexec (2.3) et d'options SAS (2.4) pouvant être invoquées dans un programme.

La maîtrise de cet environnement permet à chacun d'appréhender un environnement complexe.

Le schéma ci-dessous illustre les grandes lignes de cette partie théorique.

Les fruits, non visibles sur ce schéma, sont ce que vous réaliserez à partir de cet arbre...



2.1. Le tronc : sasv9*.cfg

2.1.1. Kesako : sasv9*.cfg

Les fichiers de configurations SAS, généralement appelés sasvX.cfg, où X prend pour valeur 8 ou 9 en fonction de la version de SAS utilisée, permettent de passer des arguments au lancement de SAS sans à avoir à les spécifier dans une ligne de commande.

Les fichiers .cfg SAS contiennent des options système sous la forme : -OPTION, ce n'est donc pas du langage SAS.

Pour savoir si une option peut être déclarée dans le sasvX.cfg, l'autoexec ou la ligne de commande de SAS, il faut consulter l'aide en ligne concernant l'option et vérifier que « Valid in : » contient « Configuration File ».

Exemple pour l'option [EMAILHOST](#).

EMAILHOST= System Option

Specifies one or more SMTP servers that supp

Valid in:	Configuration file, S
Category:	Communications: I
PROC OPTIONS GROUP=	EMAIL
Note:	This option can be

Prenons un exemple.

Pour pouvoir démarrer SAS, un espace de stockage temporaire doit être défini. Dans le jargon SAS, il s'agit de la bibliothèque Work. Sans fichier de configuration, l'option permettant de définir cette bibliothèque Work devrait être spécifiée dans la ligne de commande démarrant SAS.

Sur Unix, en supposant qu'on souhaite définir la Work dans /sas/saswork1, cette commande serait :
`sas -work /sas/saswork1 mysasfile.sas &`

On devine alors les risques d'un fonctionnement où chaque argument est placé dans la ligne de commande du programme, lors par exemple de modifications voulues par un utilisateur.

Le lancement de SAS en faisant référence au fichier de configuration se fait donc de la façon suivante :

```
sas -config /sas/install/SASFoundation/9.4/sasv9.cfg mysasfile.sas &
```

Ainsi, il n'est pas nécessaire de toucher aux arguments de lancement de SAS pour modifier des options, il suffit de modifier le fichier sasv9*.cfg qui contiendra entre autres dans notre cas la ligne :
`-work /sas/saswork1`

Pour rappel, comme pour les scripts évoqués dans le chapitre précédent, afin de conserver les modifications faites au fichier de configuration SAS en cas de mise à jour, les modifications doivent être faites dans les fichiers portant le suffixe « usermods ».

Exemple: D:\SAS\Config\Lev1\SASApp\sasv9_usermods.cfg

Les possibilités offertes par la modification des fichiers de configuration SAS sont nombreuses et permettent le moulage d'une configuration SAS répondant à tous les besoins.

En effet via la configuration SAS nous pouvons choisir la langue utilisée dans SAS (attention à l'encodage des caractères), spécifier où se trouve la SASWORK ou le profil utilisateur. Nous pouvons aussi spécifier les paramètres pour envoyer un mail à partir de SAS ou encore les paramètres d'authentification à un serveur LDAP.

Pour de plus amples informations et une liste exhaustive des possibilités offertes par les Options Systèmes SAS je vous invite à lire cette documentation : [SAS® 9.4 System Options](#)

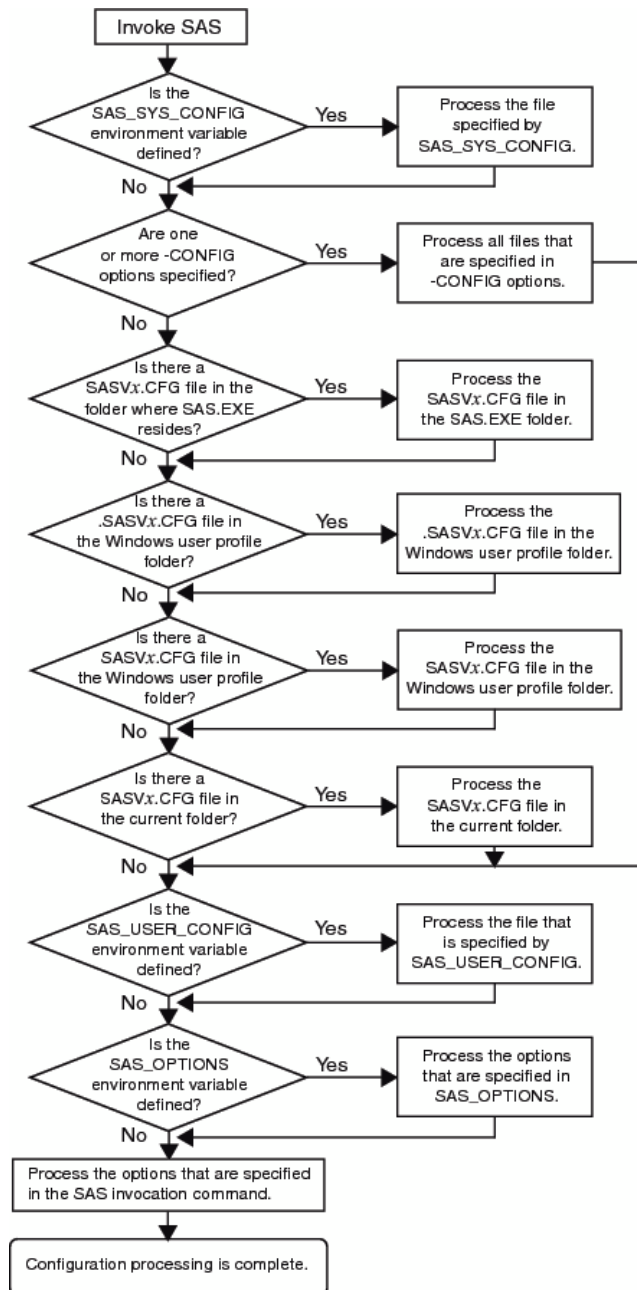
2.1.2. A plusieurs niveaux

Alors que l'on commence à voir les possibilités offertes par l'utilisation d'un fichier de configuration, on en voit rapidement les limites, un seul fichier semble bien restrictif dans le cadre par exemple d'une utilisation d'une plateforme par des milliers d'utilisateurs.

Heureusement, comme pour les scripts systèmes de configuration, les fichiers de configuration SAS fonctionnent aussi en cascade.

Le schéma ci-après reprend de façon exhaustive le cheminement de SAS quant à sa configuration.

Il est extrait de la documentation [SAS® 9.4 Companion for Windows - Files Used by SAS](#)



Il est à noter que, par défaut, dans le sasv9.cfg est appelé le fichier usermods mentionné plus haut grâce à l'option « -config », par exemple :

```
-config /sas/config/Lev1/SASApp/sasv9_usermods.cfg
```

Grâce à cet argument dans une configuration plateforme, on peut très bien imaginer le fichier sasv9.cfg d'un serveur SASApp (/sas/config/Lev1/SASApp/sasv9.cfg) faisant référence au fichier de configuration SAS créé à l'installation dans SASHOME, qui fait lui-même appel au fichier de configuration de langue défini à l'installation, puis en appelant le fichier avec le suffixe « usermods », pour enfin faire référence aux branches, l'autoexec*.sas.

2.1. L'automatisation : Autoexec.sas

2.1.1. Présentation

Le fichier autoexec SAS peut contenir des déclarations SAS qui sont exécutées immédiatement après l'initialisation de SAS et juste avant toute déclaration utilisateur.

Ces déclarations SAS peuvent être utilisées pour appeler de façon automatique un programme SAS, définir des variables qui seront utilisées pendant la session SAS, définir des options systèmes SAS, ou soumettre toute instruction SAS Base voulue.

L'autoexec n'est pas nécessaire à l'utilisation de SAS, mais automatiquement et systématiquement, au démarrage, SAS Foundation va rechercher l'autoexec.sas dans les endroits et l'ordre suivants :

- 1) Le dossier courant
- 2) Dans les dossiers spécifiés dans PATH
- 3) Dans le répertoire racine du disque courant
- 4) Dans le répertoire qui contient le binaire SAS

Si aucun autoexec.sas n'est trouvé dans ces répertoires, qu'aucun argument autoexec n'a été fourni au lancement de SAS et qu'aucune option -autoexec n'est présente dans le fichier sasv9.cfg alors aucun autoexec ne sera lancé.

2.1.2. Possibilités

Ci-après un aperçu des possibilités offertes par l'utilisation d'un autoexec.sas. Il est possible :

- de configurer une session SAS Connect via un filename rlink :

```
filename rlink  
'c:\program files\SASHome\SASFoundation\9.4\connect\saslink\tcpunix.scr';
```
- Ou de spécifier des options SAS par défaut comme les options de débogage (mprint, symbolgen, etc.) :

```
Options mprint symbolgen ;
```
- De référencer automatiquement une bibliothèque SAS :

```
Libname encours 'c:\work\sas\encours' ;
```
- D'appeler une proc format :

```
proc format library=encours;  
run;
```
- De définir le chemin de recherche des macros appelées dans la session :

```
options mautosource sasautos = ('c:\work\sas\projet1\sasmacro'  
'c:\work\sas\library\macros');
```
- Ou tout autre code SAS que vous jugez utile (définition de variable pour un environnement (prd, qual, dev)) :

```
%global env;  
%let env='Prd';  
%include '/sas/config/Lev3/SASApp/InitEnv&env.sas';
```

On aperçoit alors l'étendue de la puissance d'un autoexec dans un travail en équipe si ce dernier est utilisé par tous les membres d'une même équipe.

2.1.3. Pièges

A cause des nombreuses possibilités offertes par l'autoexec il faut aussi faire attention aux différents pièges possibles.

En effet, l'autoexec est aussi appelé en cascade, en conséquent le dernier appelé l'emportera en cas de conflit. Ceci pouvant amener des situations complexes ou des comportements non expliqués.

Il est alors recommandé de faire attention lors de la définition de bibliothèque par exemple dans l'autoexec du SASApp, de ne pas utiliser les mêmes noms sur les différents niveaux de la plateforme et d'éviter dans la mesure du possible les termes génériques anglais. Quelques exemples sont : admin, sashelp, sasuser et saswork.

Enfin, il faut faire attention à la longueur du code qui sera appelé à chaque démarrage. En effet, plus le code à traiter sera long ou plus les instructions seront complexes à traiter, plus le temps de démarrage de SAS sera long.

2.2. Bibliothèque Sasuser

2.2.1. Présentation

A chaque nouvelle session SAS, SAS vérifie la bibliothèque « sasuser » à la recherche de plusieurs éléments :

- du catalogue de Profile (sasuser.profile).
- du catalogue de registre (sasuser.registry)

La bibliothèque « sasuser » contient la définition du profil de démarrage de votre session SAS, incluant les raccourcis claviers, la configuration de l'affichage et toute autre personnalisation effectuée au cours d'une session.

Afin de savoir où se trouve votre profil vous pouvez exécuter le code suivant :

```
libname sasuser list ;
```

Si aucun profil n'est trouvé, SAS en crée un avec la configuration des raccourcis et de l'affichage par défaut.

Si SAS ne trouve pas de profil utilisateur ou si ce dernier est corrompu il essaiera dans un premier temps d'utiliser sasuser.proback, en cas d'échec et si ce dernier a accès à sashelp.profile il le copiera dans sasuser.profile avant le lancement de la session.

Il arrive qu'un profil soit corrompu, cela peut être le cas si la session SAS est quittée par la croix (X) et non par le menu ou par l'instruction « endsas ; » ou si un problème machine venait à survenir (disque plein par exemple), il est alors recommandé de créer un nouveau profil comme indiqué ci-après :

[Usage Note 43100: Possible reasons for a corrupted SASUSER profile catalog](#)

Ou ci-après, cette note nous rappelle aussi la raison d'un message d'erreur très commun quand on essaye de lancer plusieurs sessions en même temps :

[Usage Note 13838: "NOTE: Unable to open SASUSER.REGISTRY. WORK.REGISTRY will be opened instead" message when starting SAS](#)

Si le profil n'existe pas, SAS utilisera l'option système SASUSER qui peut être définie dans le sasv9.cfg de la manière suivante : -sasuser ~/sasuser

Par défaut, en 9.4, le registre utilisateur se trouve sous Windows dans :

```
C:\Users\<UTILISATEUR>\Documents\My SAS Files\9.4
```

Sous Unix dans : /home/<UTILISATEUR>/sasuser.v94

Il est à noter qu'un profil ne peut être utilisé plusieurs fois en même temps à moins de spécifier « rsasuser » au lieu de « sasuser » dans le fichier sasv9.cfg pour spécifier que le profil sera en lecture seule. Pour de plus amples informations sur ce point vous pouvez lire :

[Usage Note 13803: "NOTE: Unable to open SASUSER.PROFILE. WORK.PROFILE will be opened instead"](#)

2.2.2. Possibilités

Concrètement il est possible d'utiliser le catalogue utilisateur pour effectuer les tâches suivantes :

- Définition de raccourcis clavier,
- Définition de la taille et position des différentes fenêtres de SAS,
- Personnalisation de l'interface en ajoutant ou supprimant des éléments,
- Pré-affectation de bibliothèques quand celles-ci sont déclarées via la souris et non par une instruction libname,
- stockage de formats ou d'informats,
- stockage de macros,
- stockage de programmes SAS dans des entrées de catalogue .source, que l'on peut ensuite inclure,
- configuration de sortie de graphiques générés avec le module SAS/GRAPH (GTL).

On peut donc combiner un autoexec et son profil « personnel », telles des feuilles pour définir un environnement de travail optimal.

Pour de plus amples informations sur le sujet, je vous invite à consulter la rubrique suivante [Sasuser Library](#).

Vous pouvez aussi lire [Customizing your SAS Session](#).

2.3. SAS Options

2.3.1. Présentations

Comme évoqué précédemment, les options SAS sont le dernier niveau de personnalisation de son environnement SAS.

Lors de la spécification d'options SAS, les paramètres résultants sont appliqués à partir de la déclaration suivante jusqu'à la fin de la session SAS ou jusqu'à une nouvelle définition de la valeur.

Beaucoup d'options SAS partagent la même syntaxe que ce soit au niveau de la session SAS, ou au niveau d'une table SAS :

```
options XXX=valeur ;  
Data Ds1 ;  
  Set ds2 (XXX=valeur) ;  
Run ;
```

On peut par exemple utiliser l'option SAS "obs" pour définir le nombre d'observations sur lesquelles les traitements SAS seront exécutés, ceci dans un but de développement.

Il est aussi possible de définir des options qui permettent de rediriger une copie du journal d'exécution (altlog) ou de la sortie (altprint) dans un fichier texte ou encore d'afficher plus d'informations dans la log (mprint, mlogic, symbolgen, etc.).

Les options SAS peuvent en général être définies dans le sasv9.cfg, dans un autoexec ou tout simplement par une instruction.

Si jamais une option SAS est définie avec une faute de frappe vous obtiendrez un message du type :
ERROR: Unrecognized SAS option group name, XXXXXX.

2.3.2. Sauvegarde et restauration d'une liste d'options

La mise en place d'options afin de créer un environnement de travail peut être redondant. Si nous ne l'avons pas spécifié dans le sasv9.cfg ou dans l'autoexec nous pouvons, grâce aux procédures OPTSAVE et OPTLOAD, sauvegarder et restaurer une liste d'options dans une clef de registres SAS de la SASUSER ou dans une table SAS.

```
libname mysas "c:\sasuser\";
proc optsave out=mysas.options;
run;
proc optload data=mysas.options;
run;
```

Il est à noter que certaines options ne peuvent pas être sauvegardées. Pour savoir si cela concerne une des options utilisées, vous pouvez utiliser le code ci-dessous.

Dans un premier temps il sauvegarde les options en cours d'utilisation dans une table SAS de votre sasuser, puis cette table est affichée :

```
proc optsave;
run;
proc print;
    var optname;
run;
```

2.3.3. Restriction d'options

Pour diverses raisons, comme la bonne gestion des ressources de la machine hébergeant SAS, il peut parfois être nécessaire de restreindre les options utilisables par les utilisateurs SAS, ceci dans un but d'interdire l'exécution de binaires à partir de SAS, d'afficher le code source des macros, de redéfinir la mémoire qui peut être utilisée par SAS, etc.

Le code suivant vous donnera la liste de tout ce qui peut être restreint :

```
proc options listrestrict ;
run ;
```

et celui-ci de ce qui a été effectivement restreint sur votre plateforme.

```
proc options restrict;
run;
```

En tant qu'administrateur SAS, la modification s'effectue sous Windows dans :

<SASHOME>\rstropts\rsasv9.cfg

Sous Unix dans

<SASHOME>/misc/rstropts/rsasv9.cfg

Attention ce fichier, quand il est créé, ne peut rester vide.

Pour de plus amples informations sur le sujet, une FAQ est disponible sur le site du support :

[Comment restreindre l'utilisation d'options SAS par groupe, par utilisateur ou globalement ?](#)

3. ETUDES DE CAS :

Ce dernier point va à présent illustrer ce qui a été décrit dans les parties précédentes.

3.1. Montée de version du JRE

Lors du téléchargement ou de la réception d'un dépôt, ce dernier contient toujours le JRE minimum recommandé (Private JRE). Seulement, il arrive qu'Oracle, IBM et les différents éditeurs de JRE doivent combler certaines failles de sécurité notamment.

Il arrive donc dans la vie du logiciel SAS qu'un utilisateur ou un administrateur doive mettre à jour son JRE.

1. Dans un premier temps, avant de faire quoi que ce soit, il faut se renseigner sur la version du JRE supportée par la version utilisée de SAS.
Par exemple, en 9.2 TS2M3 sur une plateforme Linux on peut voir [ici](#) que la version 1.5u14 ou 1.6u14 est supportée.
Pour information, le support d'une plateforme ou d'un environnement SAS est assuré à partir du moment où la mise à jour est faite vers une version mineure.
Concrètement pour notre plateforme en exemple nous pouvons monter jusqu'à la version 1.6u45 qui est et sera la dernière version du JRE en version majeure 1.6.
2. Ensuite il faut aller sur le site d'Oracle pour télécharger la version retenue dans la plupart des cas il s'agira de la version 32 bits.
3. Puis il faudra copier le dossier des polices fournies avec SAS dans le dossier de la nouvelle JRE, les polices utilisées par les clients SAS sont présentes (sous Unix ou sous Windows) dans :
`$SASHOME/jre-installed-version/lib/fonts`
4. Pour que cela soit pris en compte dans SAS Foundation sous UNIX, il s'agit de modifier le fichier SASENV afin de faire évoluer JAVA_HOME vers votre répertoire d'installation du JRE mis à jour.
Sous Windows la variable JAVA_HOME est mise à jour par le programme d'installation.
5. Sous Windows uniquement, il faut modifier le fichier sasv9.cfg se trouvant dans le chemin ci-après, pour que cela soit pris en compte notamment par le module SAS/Graph. Il est à noter que le langage peut correspondre par exemple à 'fr' ou 'en' par exemple, le chemin est à adapter selon votre configuration :
`!SASROOT\nls\language\sasv9.cfg`
La modification du fichier est nécessaire pour mettre à jour le paramètre -JREOPTIONS et plus précisément le sous-paramètre : -Dsas.jre.libjvm
6. Afin que cela soit pris en compte pour les clients tel que SAS Management Console, il faut mettre à jour la variable JREHOME dans le fichier : `$SASHOME/sassw.config`.
7. Afin que cela soit pris en compte dans une configuration LevX, il faut mettre à jour la variable JAVA_JRE_COMMAND dans le fichier level_env.bat ou level_env.sh à la racine de votre configuration LevX.

Pour plus de plus amples informations ou mettre à jour un JDK, vous pouvez consulter le lien suivant : [Configuring SAS 9.3 to Use an Alternative Java Runtime Environment or Java Development Kit](#)

3.2. Montée de version d'un client SGBD et plus précisément Oracle

Il peut parfois aussi être nécessaire de mettre à jour son client base de données lorsque l'on possède une licence de type SAS/Access, l'exemple ci-après reprendra le cas du client Oracle.

1. Dans un premier temps il s'agit de vérifier que le client dans la version choisie est bien compatible avec la version du serveur de base de données.
En effet, il serait risqué d'utiliser un client Oracle 11G sur une base Oracle 9 (ou l'inverse).
2. Dans un second temps, il faut s'assurer que le client choisi bénéficie de la même architecture que sa plateforme SAS et en général de son système d'exploitation, le choix se fait entre x86 (32 bit) et x64 (64 bit).
3. Une fois le client installé, il est fortement conseillé de tester une connexion à votre base préférée (sans passer par SAS) afin de vérifier que le client fonctionne comme attendu et que le fichier contenant les connexions tel que le tnsname.ora pour oracle a bien été mis à jour.
4. Ensuite, à l'aide du guide qui vous a servi à l'installation / configuration qui se nomme « Configuration Guide for SAS 9.x Foundation for XXX » où x correspond à la version de SAS et XXX correspond au système d'exploitation, dans la partie SAS/Access qui vous concerne sont renseignées les variables d'environnement qu'il faut mettre à jour.
Sous les systèmes UNIX, il s'agit soit de la variable LIBPATH ou SHLIBPATH ou LD_LIBRARY_PATH ou encore LD_LIBRARY_PATH64.
Sous Windows, il s'agit du PATH.
5. Enfin, sous Windows comme sous Unix, il ne faut pas oublier de recréer le lien via SAS Deployment Manager en 9.3 et 9.4



Pour les versions précédentes (9.2 et 9.1.3), la configuration s'opère en lançant le sassetup (\$SASROOT/sassetup) dans la partie : Perform Product-Specific Configuration.

```
* 1. Run Setup Utilities
-----
g: Go back   q: Quit   h: Help
-----
Action? (1) 1

Setup Utilities Menu
-----
* 1. Renew SAS Software
2. Perform SAS Software Configuration
3. Perform Product-Specific Configuration
4. Run Technical Support Tools
5. List Previously Installed Software
-----
g: Go back   q: Quit   h: Help
-----
Action? (1) 3

Product-Specific Configuration Menu
-----
* 1. SAS/ACCESS Configuration
2. Desktop Configuration
3. Update SAS/IntrNet Services Configuration Program
4. Run Configuration Steps for Risk Dimensions
-----
g: Go back   q: Quit   h: Help
-----
Action? (1) 1

SAS/ACCESS Software Configuration

Several SAS/ACCESS products require post-installation configuration
before you can use the software. Part of this configuration
is identifying the DBMS version you are running for the
selected SAS/ACCESS software products.

You have the option of performing this configuration step during
installation or at a later time after the installation is complete.

If you choose to configure SAS/ACCESS software during installation
(which is recommended), select the SAS/ACCESS software to configure
from the list that follows and provide answers to the additional
questions when presented.

When your installation is complete, refer to the Configuration
Guide for details on how to complete your SAS/ACCESS software
configuration.

Would you like to configure your SAS/ACCESS software at this time? (Y)
-->Y
----- ACCESS Config Selection Screen ----- Page 1 of 1
'*' = selected
-----

1  SAS/ACCESS DB2                      2  SAS/ACCESS INFORMIX
3  SAS/ACCESS Interface to Greenplum  4  SAS/ACCESS MS SQL Server
```


3.3. Changement de localisation de la WORK

A chaque démarrage d'une session SAS, une bibliothèque temporaire est créée : WORK.

1. Il est possible de déterminer où elle se trouve dans une session en soumettant le code suivant :
`libname work list;`
2. Il est possible de définir le chemin où sera créée la bibliothèque à chaque session en incluant dans le `sasv9.cfg` l'argument WORK. Voici une illustration, sous un système de type Unix, pour définir la work dans le répertoire `/saswork/new/` :
`-work /saswork/new/`
3. Il est aussi possible d'utiliser une variable d'environnement au sein de la définition de la bibliothèque Work. Prenons un exemple sous Windows : la variable d'environnement `mavar` est d'abord déclarée, puis est utilisée (notez le caractère « ! » pour la référencer). La bibliothèque WORK est ainsi créée dans `c:\tempdir\WORKSAS\` :
`-set mavar c:\tempdir`
`-work !mavar\WORKSAS\`
4. La définition de la variable n'est pas obligatoirement faite dans le `sasv9.cfg`. Elle peut être créée côté système dans une variable Unix qui porterait par exemple le nom de `$mavar` ou sous Windows `%MAVAR%`.
5. Enfin, nous pouvons faire référence à un fichier qui contiendrait plusieurs répertoires. Il suffirait alors d'y faire référence comme ceci :
`-work /sas/workfile.txt`

Ce fichier peut alors contenir autant de lignes que voulues afin de répartir la work sur autant de disques voulus et ceci dans un souci de gestion d'espace ou dans un souci de gestion des entrées/sorties.

Le fichier `workfile.txt` pourrait alors ressembler à ceci pour optimiser les entrées/sorties :

```
/disk1/saswork
/disk2/saswork
/disk3/saswork
method=random
```

Si l'on voulait optimiser la gestion de l'espace disque il faudrait remplacer `method=random` par : `method=space`

Pour plus de détails sur le sujet vous pouvez lire [Gestion de la bibliothèque WORK](#).

3.4. Changement des paramètres e-mail

SAS, que cela soit Foundation ou Plateforme, permet d'envoyer des e-mails. Aujourd'hui la plateforme Windows offre une plus grande flexibilité en termes de choix.

En effet SAS est compatible avec 3 « protocoles » :

- MAPI (messaging application programming interface) qui est une interface de programmation de messagerie et a été développé par Microsoft. Ce dernier permet entre autres d'envoyer des e-mails via Microsoft Outlook si ce dernier est installé sur la machine où SAS se trouve.
- Son concurrent, VIM (Vendor Independant Messaging) qui était une API proposée au début des années 90 par Borland, IBM, Novell et Lotus. Ce dernier permet entre autres d'envoyer des e-mails via Lotus Notes si ce dernier est installé sur la machine où SAS se trouve.
- SMTP (Simple Mail Transfer Protocol) qui est aujourd'hui le protocole mail le plus utilisé notamment en entreprise pour des applications tournant sous Unix par exemple.

Quel que soit le choix effectué, le principe reste le même.

Il faut définir plusieurs paramètres dans le fichier sasv9.cfg :

1. -EMAILSYS qui sera défini à MAPI, VIM ou SMTP
2. -EMAILHOST qui contiendra le serveur de messagerie SMTP
3. -EMAILID qui contiendra le nom d'utilisateur, le nom du profil ou l'adresse e-mail
4. -EMAILPORT qui peut contenir le port du serveur SMTP si ce dernier est différent de 25
5. -EMAILPW qui va de pair avec EMAILID et qui est le mot de passe associé. Il peut être encrypté ou pas et entre quote si le mot de passe contient un espace.

Comme expliqué dans le point 2.1.1, il est possible de spécifier ces arguments dans la ligne de commande du client SAS ou du serveur SAS.

Ci-après quelques exemples,

- Pour une configuration avec Lotus Notes :
-emailsys vim
-emailid "Paul Adam"
-emailpw 'mdp34\$ email'
- Pour une configuration avec Outlook :
Il faut juste s'assurer que le paramètre emailsys soit par défaut à MAPI, ce qui peut être vérifié avec le code suivant :
proc options group=email;
run;
- Enfin pour une configuration SMTP :
-emailsys SMTP
-emailhost serveursmtp.sas.com
-emailport 253
-emailid youness.sammour@sas.com
-emailpw {SAS002}DBCC571245AD0B31433834F80BD2B99E16B3C969

Il devient alors possible d'envoyer un mail à Nicolas Dupont avec le code suivant :

```
filename monmail email "nicolas.dupont@sas.com" subject="TEST e-mail";  
data _null_;  
  file monmail;  
  put 'Bonjour le Monde';  
run;
```

Pour de plus amples informations ou plus de détails, vous pouvez lire la SAS NOTE 19767 [Using the SAS System to send SMTP e-mail](#) ainsi que la documentation Windows [EMAILSYS System Option](#) puis la documentation [Sending E-mail using SAS](#).

3.5. Définition d'un autoexec

Il est parfois un peu fastidieux de définir les mêmes variables et les mêmes bibliothèques à chaque nouvelle session, que cela soit dans un contexte SAS Foundation ou sur une plateforme. Comme évoqué plus haut l'autoexec permet d'automatiser des actions SAS à chaque lancement du serveur Workspace (et donc au lancement des sessions SAS Enterprise Guide), du serveur Stored Process, du serveur OLAP, du serveur Connect ainsi que tout autre serveur logique SAS.

- Le comportement évoqué en 2.1.1 est général à SAS Foundation, s'il n'est pas spécifié via l'option -autoexec dans le sasv9.cfg, l'autoexec est cherché dans le dossier courant, dans les dossiers spécifiés dans le PATH, dans le répertoire racine du disque courant et dans le répertoire qui contient le binaire SAS.
- Afin de définir un autoexec au lancement de SAS Enterprise Guide sans avoir à le définir dans le précode du client, il est possible d'utiliser le fichier autoexec_usermods.sas qui se trouve dans le répertoire <config>\Lev1\SASApp\WorkspaceServer. Ce dernier est appelé par l'autoexec.sas présent dans le même répertoire.

Exemple de fichier autoexec.sas :

```
Options nomacrogen NoSymbolgen nomlogic nomprint nomfile;

libname project1 'c:\work\sas\library';
libname project2 'c:\work\sas\projet1';

options mautosource sasautos =
('c:\work\sas\projet1\sasmacro'
'c:\work\sas\library\macros');

%global env;
%let env='Prd';
%include '/sas/config/Lev3/SASApp/InitEnv&env.sas';
```

4. EN CAS DE PROBLÈME

Si vous rencontrez des problèmes d'environnement ou de configuration, vous pouvez nous écrire à support@sas.com, en attachant à votre message, un document explicitant votre problème, ce dernier accompagné des fichiers de configurations (sasvX.cfg, autoexec et profil utilisateur) et de la sortie de la commande « env » sous Unix ou « set » sous Windows (pour l'utilisateur SAS).

5. LIENS UTILES

Je vous recommande tout d'abord les guides par systèmes d'exploitation :

- [SAS® 9.4 Companion for Windows](#)
- [SAS® 9.4 Companion for Unix](#)
- [SAS® 9.4 Companion for z/OS](#)

Concernant la bibliothèque SASUSER, les notes suivantes sont les plus utiles en cas de problème :

- [Usage Note 43100: Possible reasons for a corrupted SASUSER profile catalog](#)
- [Usage Note 13838: NOTE: Unable to open SASUSER.REGSTRY. WORK.REGSTRY will be opened instead message when starting SAS](#)
- [Usage Note 13803: "NOTE: Unable to open SASUSER.PROFILE. WORK.PROFILE will be opened instead"](#)

Les liens ci-après vous renseigneront dans les moindres détails sur ce qu'il est possible de faire pour personnaliser sa session SAS :

- [Customizing your SAS Session](#)
- [Working within Your SAS Session](#)
- [Sasuser Library](#)
- [Profile Catalog](#)

Ci-après quelques articles écrits par le support SAS France sur des sujets évoqués précédemment :

- [Gestion de la bibliothèque WORK.](#)
- [Comment changer la langue des clients JAVA disponibles avec SAS® sous Windows ?](#)
- [Comment restreindre l'utilisation d'option SAS par groupe, par utilisateur ou globalement ?](#)

Enfin, les quelques documentations issues des « manuels » évoquées dans l'introduction théorique puis dans les exemples :

- [SAS® 9.4 System Options](#)
- [SAS® 9.4 Companion for Windows - Files Used by SAS](#)
- [Using the SAS System to send SMTP e-mail](#)
- [EMAILSYS System Option](#)
- [EMAILHOST](#)
- [Configuring SAS 9.3 to Use an Alternative Java Runtime Environment or Java Development Kit](#)

6. CONCLUSION

Le sujet de SAS et son environnement est un sujet vaste et complexe. J'espère que cet article aura su vous faciliter l'appréhension de cette base nécessaire au bon fonctionnement de SAS.

Si vous avez des questions sur le sujet, qu'elles soient théoriques ou pratiques, le support reste à votre disposition pour en discuter.

Younes SAMMOUR
Consultant Support Clients SAS France