



## SAS ET ODBC, MAIN DANS LA MAIN POUR ALLER PLUS LOIN

SAS® intègre la possibilité de se connecter à une large gamme de sources de données. Ces sources de données sont implémentées en tant que connexions natives, via les modules **SAS/ACCESS® Interface to**. Ainsi, SAS dispose de techniques et d'optimisations spécifiques à ces sources de données.

### Caractéristiques :

Catégories : SAS  
OS : Windows, Unix  
Version : SAS® 9.4M3  
Vérifié en Novembre 2016

Par l'intermédiaire du module **SAS/ACCESS Interface to ODBC**, SAS permet d'utiliser la norme ODBC générique pour l'accès aux sources de données extérieures. Les avantages de la norme ODBC ne sont plus à démontrer. De nombreux fournisseurs de bases de données mettent des pilotes à la disposition des utilisateurs pour la connexion à leurs bases de données à partir de tous les logiciels permettant une connexion vers une source ODBC. **SAS/ACCESS Interface to ODBC** offre la possibilité d'utiliser ces pilotes ODBC pour se connecter aux données à partir de SAS. Ainsi, les développeurs SAS n'ont pas besoin de modifier leurs programmes pour leur permettre d'accéder à un serveur de données spécifique. Il suffit que le pilote propre à la base de données soit correctement configuré.

Cet article est une présentation de la norme ODBC, du module **SAS/ACCESS Interface to ODBC**, de son installation et de son utilisation dans SAS. Je vous présente également des astuces, les réponses aux questions les plus souvent posées au support SAS France ainsi que les solutions aux problèmes fréquemment rencontrés.

Cet article est écrit comme un document dans lequel vous pouvez venir piocher des informations, astuces et tutoriaux dans le but de vous familiariser et de vous accompagner au quotidien dans l'utilisation d'ODBC.

Je ne pouvais pas terminer cette introduction sans vous rappeler que SAS utilise également cette norme ODBC pour permettre à des applications tierces d'accéder et de lire des données SAS. Cette partie sera abordée dans un prochain article.

## Sommaire

1.	Qu'est-ce qu'ODBC ? .....	3
2.	SAS et ODBC.....	3
3.	SAS/ACCESS Interface to ODBC ou comment accéder depuis SAS aux données d'une base tierce 4	
3.1.	Schéma de fonctionnement.....	5
3.2.	Présentation de l'architecture .....	6
3.3.	Configuration de SAS/ACCESS to ODBC.....	6
3.4.	Est-ce que les pilotes ODBC sont validés par SAS ? .....	7
3.5.	Pilote 32 bits ou 64 bits ? .....	8
3.6.	Ajouter le module SAS/ACCESS Interface to ODBC à une installation SAS existante .....	10
3.7.	Configuration pas à pas sous Windows .....	11
3.8.	Configuration pas à pas sous Unix.....	12
3.8.1.	Configuration du fichier ODBC.ini .....	12
3.8.2.	Configuration SAS.....	13
3.9.	Check-list des tâches à réaliser avant validation .....	14
3.10.	Validation de la connexion à ODBC, dans une session SAS .....	15
3.11.	Création d'une bibliothèque ODBC dans les Métadonnées.....	17
3.12.	Se connecter à une base sans configurer de DSN ODBC.....	22
3.13.	Faire cohabiter SAS/ACCESS to SQL Server et SAS/ACCESS Interface to ODBC sous Unix 22	
3.14.	La boîte à outils SAS/ACCESS to ODBC.....	23
3.14.1.	Définition d'un schéma lors d'une connexion via SAS/ACCESS to ODBC .....	24
3.14.2.	Gestion des synonymes Oracle .....	27
3.14.3.	Travailler avec des caractères spéciaux et des noms de plus de 32 caractères. ....	28
3.15.	Quelle est la différence entre ODBC et OLE DB ?.....	30
3.16.	Quelques problèmes connus et limitations avec SAS/ACCESS to ODBC .....	31
3.16.1.	Problèmes lors de la création de la bibliothèque (via une instruction LIBNAME) .....	31
3.16.2.	Problèmes pouvant survenir lors de la manipulation de données .....	32
4.	Contacter le support SAS en cas de problème.....	34
5.	Liens utiles.....	34
6.	Conclusion .....	35

## 1. QU'EST-CE QU'ODBC ?

---

ODBC (*Open DataBase Connectivity*) est une interface de programmation (API) universelle, indépendante de la source de données. Créé au début des années 90 par Microsoft, ODBC a tout de suite eu un objectif très ambitieux : permettre aux utilisateurs d'accéder facilement à des données de toute base de données relationnelle en utilisant une interface commune. Les applications utilisent ainsi les mêmes fonctions de l'API quelle que soit la source de donnée ciblée. Dans ce cas, le pilote se charge de mener à bien l'action demandée, en fonction du type de source de données. Dès sa conception, ODBC a été destiné à devenir la norme pour l'accès universel aux données.

Il est aussi disponible sur les systèmes d'exploitation Unix. Beaucoup de SGBD adhèrent à ce standard, permettant à ses utilisateurs d'accéder aux données créées par d'autres logiciels.

## 2. SAS ET ODBC

---

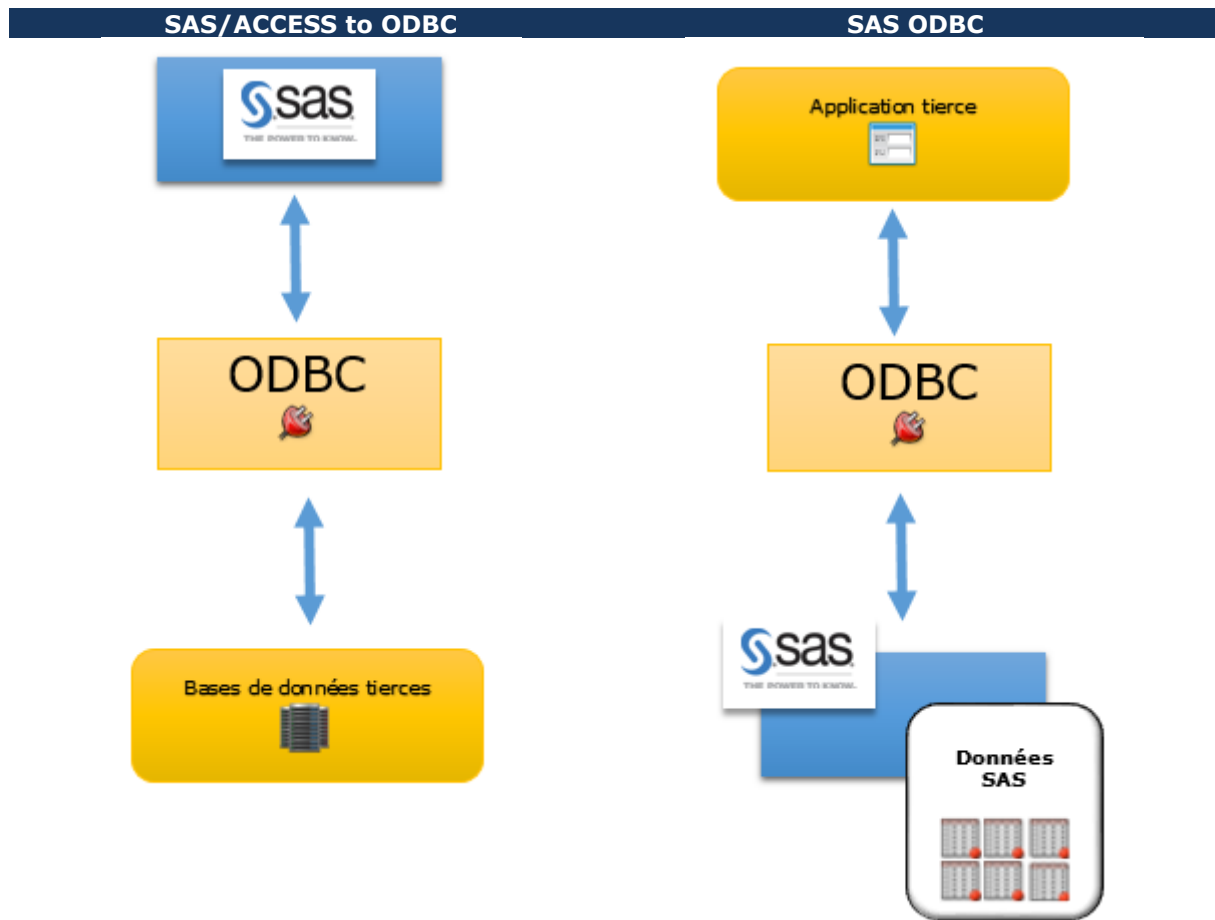
Comme indiqué en introduction, deux scénarios impliquent le Système SAS et ODBC.

Dans le premier cas, le Système SAS est l'application cliente. L'utilisateur SAS veut accéder à des données non SAS. Il doit avoir à disposition le driver ODBC du produit auquel il veut accéder. Au niveau du Système SAS, il doit avoir le module SAS/ACCESS Interface to ODBC.

*Exemple : un utilisateur SAS veut pouvoir accéder, depuis le Système SAS, à des données SQL Server. Il lui faut le un driver ODBC SQL Server et le module SAS/ACCESS Interface to ODBC.*

Dans le deuxième cas, le Système SAS est le serveur de données. L'utilisateur non SAS veut accéder à des données SAS. Il doit avoir à disposition le driver ODBC SAS (fourni par SAS) et bien-sûr une application cliente adhérant au standard ODBC.

*Exemple : un utilisateur MS-EXCEL veut pouvoir accéder, depuis MS-QUERY, à des données SAS. Il lui faut le driver ODBC SAS et bien-sûr un Système SAS quelque part sur la même machine ou ailleurs sur un réseau.*



### 3. SAS/ACCESS INTERFACE TO ODBC OU COMMENT ACCEDER DEPUIS SAS AUX DONNEES D'UNE BASE TIERCE

---

Une source de données ODBC doit être préalablement configurée à l'aide d'un pilote ODBC tiers. Cette source doit pouvoir être validée par une application autre que le Système SAS, c'est-à-dire testée par un autre logiciel comme par exemple MS-QUERY. Cette simple manipulation évite bien des soucis, et permet de simplifier la chaîne d'éléments présents (i.e. on vérifie d'abord que l'accès fonctionne sans SAS avant de rajouter la couche SAS).

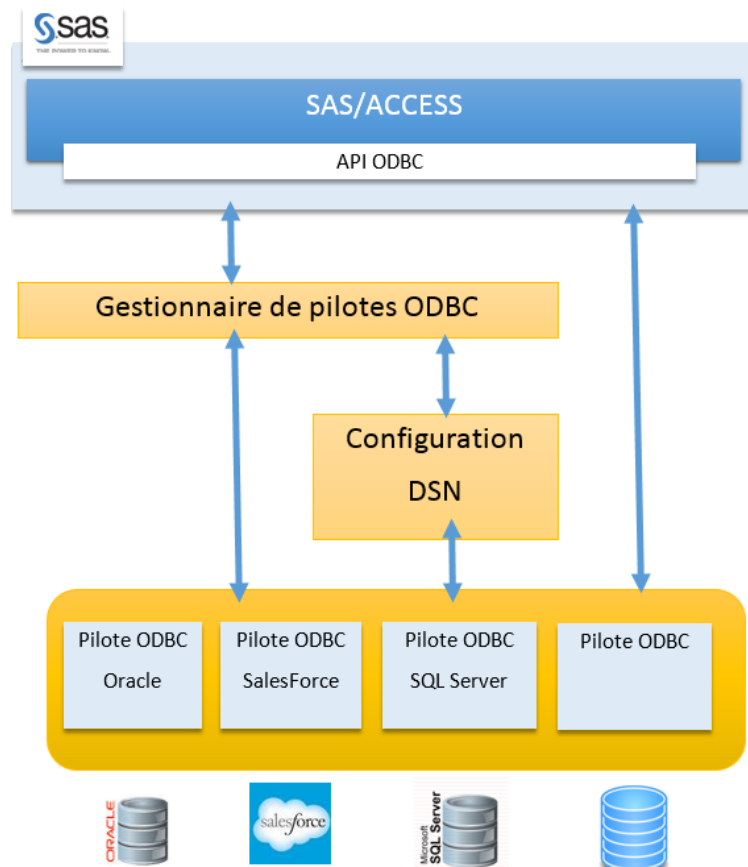
*Exemple : un utilisateur SAS veut pouvoir accéder, depuis le Système SAS, à des données MySQL. Il a configuré une source de données ODBC et veut désormais requêter cette source de données depuis le Système SAS.*

Avant cela, il faut valider l'accès à cette source de données. Si cela fonctionne, on peut alors passer au Système SAS. Si cela ne fonctionne pas, l'utilisateur doit voir avec son support informatique pour détecter l'origine du problème.

La configuration d'une source de données ODBC tierce est à la charge de l'utilisateur. Les chapitres [4.7](#) et [4.8](#) détaillent les étapes à suivre pour installer votre pilote sur votre environnement Windows ou Unix.

Une fois la source testée et validée dans un monde non SAS, on peut tenter d'y accéder depuis le Système SAS.

### 3.1. Schéma de fonctionnement



Si on se réfère à l'illustration ci-dessus, le module SAS/ACCESS utilise l'API ODBC. En tant qu'utilisateur, on pointe simplement sur la source donnée en fournissant les paramètres de connexion propres à celle-ci.

Cette tâche se fait via le gestionnaire de sources de données ODBC.

L'ensemble des paramètres de connexion se résume à un nom : celui que l'on donne à une source de données (DSN en anglais pour Data Source Name).

## 3.2. Présentation de l'architecture

L'architecture ODBC se compose de 4 composants :

- **L'application.** Le programme qui utilise l'API ODBC, soumet les requêtes SQL et lit les résultats. Dans le cas de l'utilisation du module SAS/ACCESS to ODBC, SAS est l'application.
- **Le pilote (ou driver) ODBC.** Le pilote ODBC soumet les requêtes SQL à la source de données et retourne les résultats à l'application. Le pilote adapte la syntaxe de la demande afin de se conformer à la syntaxe requise par la source de données. Vous avez donc besoin d'un pilote qui communique avec le serveur où sont stockées les données auxquelles vous souhaitez accéder. Pour faire simple, si vos données sont stockées sur une base MongoDB, vous avez besoin d'un pilote MongoDB. De nombreuses sociétés créent et fournissent des drivers ODBC. Les fournisseurs de base de données, comme Oracle ou IBM, fournissent leurs propres pilotes ODBC (gratuitement ou pas). Des sociétés sont également spécialisées dans le développement de pilotes ODBC, comme la société Datadirect (il y en a d'autres).

**Le gestionnaire ODBC.** Ce programme charge en mémoire les pilotes ODBC afin qu'ils puissent être accessibles par l'application utilisant l'API.

- **Les sources de données.** Si vous n'avez pas de sources de données, déployer une solution ODBC est inutile. Oracle, Teradata, Salesforce.com, Twitter, Microsoft SQL Server, MongoDB, HBase, ou encore Google BigQuery sont des exemples de sources de données. Ce ne sont que quelques exemples, il y a en effet beaucoup de sources de données qui peuvent être traitées en utilisant ODBC. Dans les faits, SAS peut accéder aux données de n'importe quelle base de données possédant un pilote ODBC qui répond à la norme ODBC. Cette fonctionnalité permet à l'interface SAS/ACCESS pour ODBC d'être très populaire auprès des utilisateurs de SAS.

## 3.3. Configuration de SAS/ACCESS to ODBC

La première étape est l'installation du gestionnaire de pilotes/drivers ODBC (ODBC Driver Manager). Il permet de gérer l'interaction entre l'application cliente et le(s) driver(s) ODBC. Il est disponible par défaut sur Windows. Sur Unix, il faut contacter le fournisseur de la base de données ou un fournisseur tiers (comme par exemple Datadirect / Easysoft ou unixODBC) pour récupérer les binaires et l'installer sur la machine où SAS est installé.

Concernant UnixODBC, SAS 9.4 requiert UnixODBC [version 2.3.2 au minimum](#). La documentation officielle précise également les [méthodes ODBC](#) utilisées par SAS afin de vous assurer que le gestionnaire et le pilote que vous utilisez fonctionnera dans les meilleures conditions.

Certains environnements Linux ont, bien souvent, un gestionnaire de pilotes unixODBC installé dans un emplacement global tel que /usr/lib ou /usr/lib64. Cependant, il s'agit généralement d'une version ancienne, bien souvent inférieur à la version 2.3.2 SAS ne supporte pas l'utilisation de SAS/ACCESS Interface ODBC avec ces anciennes versions du gestionnaire de pilotes unixODBC.

Ensuite, il faut aussi récupérer les bibliothèques du pilote ODBC du SGBD et l'installer sur la même machine. Il est préférable d'utiliser les pilotes ODBC développés par fournisseurs de pilotes ODBC tels que Easysoft ou DataDirect de Progress Software. Ceux-ci sont généralement plus fiables et mieux pris en charge que leur équivalent gratuit.

### 3.4. Est-ce que les pilotes ODBC sont validés par SAS ?

Au moment de choisir le pilote ODBC pour vous connecter à votre base de données via SAS/ACCESS Interface to ODBC, vous pouvez, de façon tout à fait légitime, vous poser la question si le pilote choisi a été testé avec SAS/ACCESS to ODBC.

Par exemple, vous allez connecter SAS à SAP HANA et voulez savoir si cette « association » a été testée par les équipes R&D de SAS.

Notez d'abord que, dans ce cas présent, SAS fournit un moteur SAS/ACCESS spécifique pour SAP HANA nommé SAS/ACCESS to SAP HANA. C'est la même chose avec Oracle, Sybase...

Mais qu'en est-il avec ODBC puisque le moteur SAS/ACCESS Interface to ODBC est un moteur générique ?

Il y a deux approches concernant cette question :

- Est-ce que le pilote ODBC est supporté ?
- Est-ce que le pilote ODBC a été testé ?

Dans notre cas, nous souhaitons savoir si la R&D a testé le pilote ODBC SAP HANA.

Démarrons par cette notion de « **supporté** ». Concernant le support de tel ou tel pilote ODBC, la réponse est simple :

**S'il existe un pilote ODBC compatible ODBC 3.5 pour la base de données, celui-ci est supporté par SAS/ACCESS to ODBC.**

« **Supporté** » ne signifie pas qu'il a été prouvé que le pilote fonctionne avec SAS. Cela signifie que le support technique essaiera de vous aider, en cas de problème, à faire fonctionner la connexion entre ce pilote et SAS.

J'insiste bien sur le mot « essaiera » car il peut y avoir des problèmes liés au pilote ODBC qui, malheureusement, ne peuvent pas être résolus. Cela dépend de la qualité du pilote ODBC.

Maintenant, est-ce que notre pilote ODBC SAP HANA a été « **testé** » par la R&D ?

La R&D ou le Support Clients SAS n'a pas testé tous les pilotes ODBC disponibles sur le marché. Le plus important est que votre pilote ODBC soit compatible avec la norme ODBC 3.5.

### 3.5. Pilote 32 bits ou 64 bits ?

La question risque de se poser au moment de votre installation. Dois-je utiliser un pilote (driver) 32 ou 64 bits ? La réponse est simple :

- SAS en 32 bits nécessite un pilote et un gestionnaire ODBC 32 bits.
- SAS en 64 bits nécessite un pilote et un gestionnaire ODBC 64 bits.

Pensez également que SAS en 64 bits nécessite un ODBC driver manager et un ODBC driver compilé en 64 bits (<http://support.sas.com/kb/15/615.html>)

Aussi, si vous utilisez SAS/ACCESS Interface to ODBC en version 64 bits, vous ne pouvez pas accéder à des drivers ODBC 32 bits. À ce jour, certains fournisseurs offrent deux pilotes ODBC 32 bits et 64 bits pour leurs sources de données (par exemple, Oracle, Sybase, Teradata, et Microsoft SQL Server). A noter également que votre base de données peut-être en 32 ou 64 bits, mais cela n'a pas d'importance. Seule la version du pilote utilisé compte.

Cependant, vous pourriez trouver des pilotes ODBC uniquement en 32 bits. C'est le cas, par exemple, pour les sources de données Microsoft Office tels qu'Excel et Access. (<http://support.sas.com/kb/38/779.html>)

A noter également, une version 64 bits du système d'exploitation Microsoft Windows inclut les versions suivantes de l'outil administrateur de sources de données Microsoft ODBC Open Database Connectivity (Odbcad32.exe) :

Version 32 bits	%systemdrive%\Windows\SysWoW64.
Version 64 bits	%systemdrive%\Windows\System32.

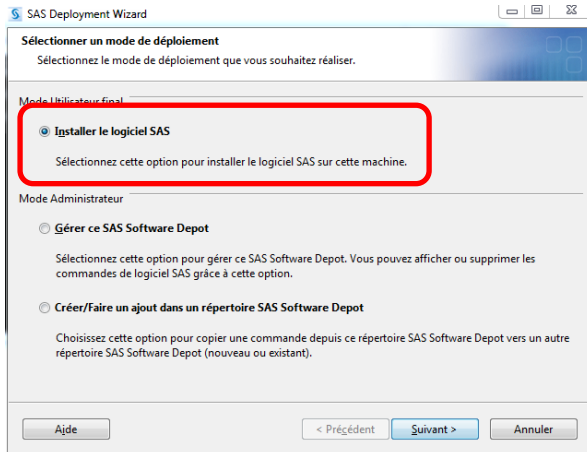
Point d'attention : l'administrateur ODBC affiche les deux versions 32 bits et 64 bits des sources de données utilisateur DSN dans une version 64 bits de Windows (<http://support.microsoft.com/fr-fr/kb/942976/fr>). La version 32 bits de l'outil Administrateur ODBC affiche les sources de données



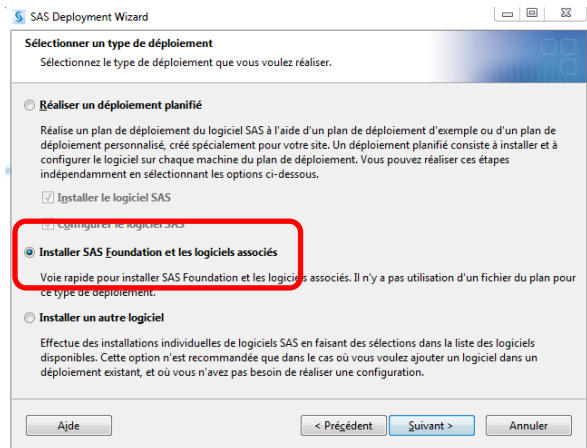
du système 32 bits, utilisateur 32 bits DSN et utilisateur 64 bits DSN. La version 64 bits de l'outil Administrateur ODBC affiche les sources de données système de 64 bits, utilisateur 32 bits DSN et utilisateur 64 bits DSN.

### 3.6. Ajouter le module SAS/ACCESS Interface to ODBC à une installation SAS existante

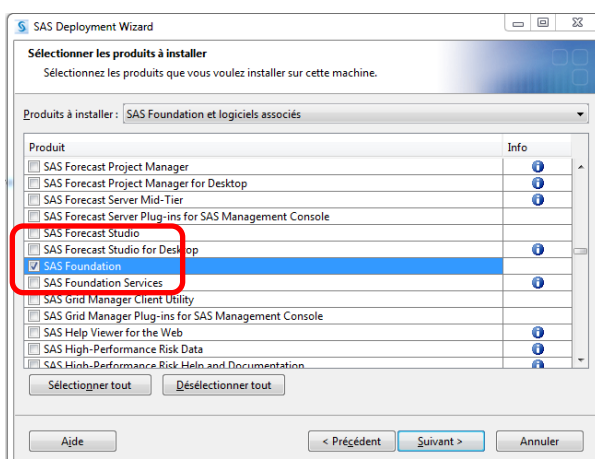
Les copies d'écran suivantes présentent les principaux écrans affichés lorsque vous souhaitez ajouter le module SAS/ACCESS Interface to ODBC à une installation SAS existante :



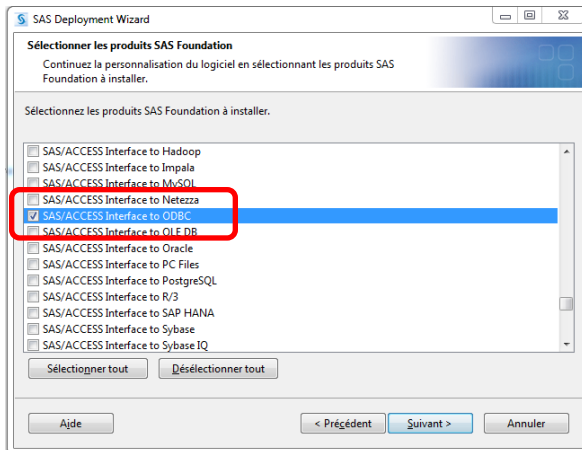
Dans **SAS Deployment Wizard**, choisissez "Installer le logiciel SAS" puis cliquez sur « Suivant »



Sélectionnez le type de déploiement en choisissant "Installer SAS Foundation et les logiciels associés »



Dans l'écran de sélection des produits à installer, sélectionner "SAS Foundation"

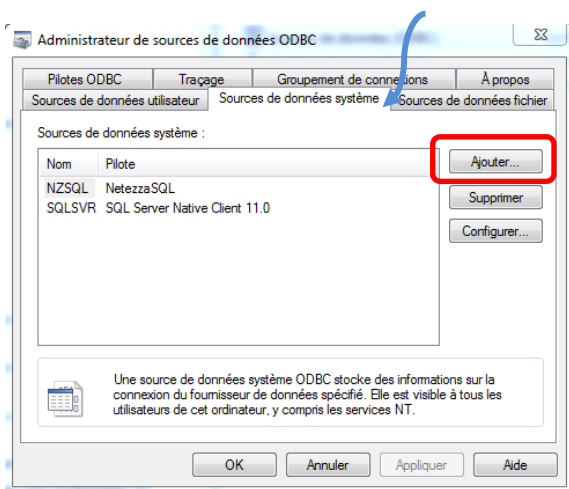


Dans l'écran de personnalisation du logiciel, sélectionnez « SAS/ACCESS interface to ODBC »

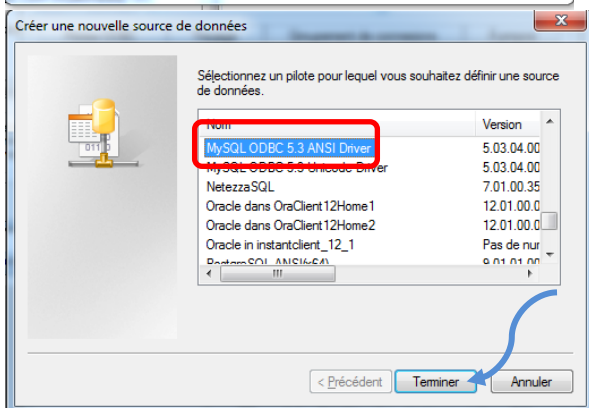
### 3.7. Configuration pas à pas sous Windows

Pour retrouver les configurations des connexions ODBC sous Windows 7, ouvrez l'administrateur de la source de données ODBC :

Panneau de configuration > Système et sécurité > Outils d'administration > Sources des données (ODBC) > Sources des données système



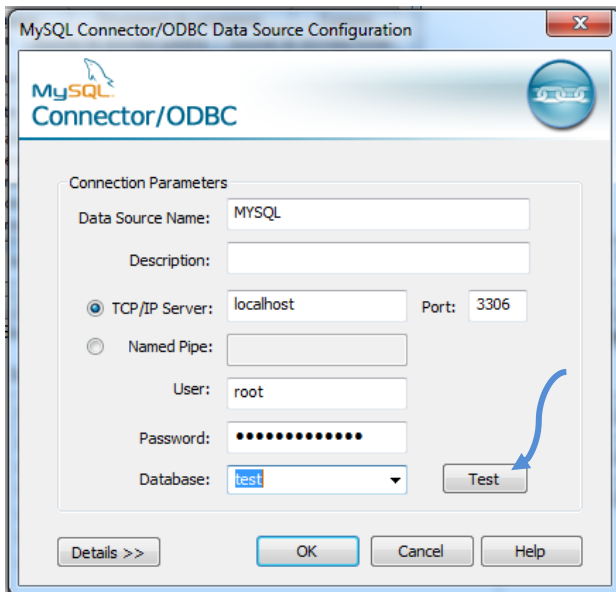
Dans l'« administrateur des sources de données ODBC », choisissez « Source de données système »



Sélectionnez un pilote pour lequel vous souhaitez définir une source de données puis cliquez sur « terminer »

Exemple : *MySQL ODBC 5.3 ANSI Driver*

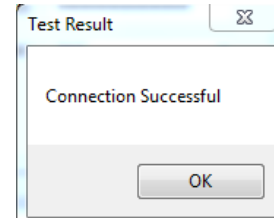
Vous devez avoir installé ce pilote au préalable.



Une fenêtre s'ouvre afin que vous puissiez définir les paramètres de connexion à votre base de données.

Chaque pilote ODBC propose un écran de configuration différent.

Une fois les paramètres saisis, vous pouvez tester votre connexion en cliquant sur le bouton « Test »



### 3.8. Configuration pas à pas sous Unix

La configuration de ODBC peut se faire manuellement, en éditant des fichiers texte `odbc.ini` ou via l'outil graphique appelé `ODBCConfig`.

#### 3.8.1. Configuration du fichier `ODBC.ini`

Dans notre exemple nous allons ajouter une DSN ODBC pour SQL Server :

```
[SQL_SERVER]
Driver=/home/sas/SQLSERVER_ODBC_DRIVER/S0sqls26.so
Description=SQL Server Wire Protocol
AnsiNPW=1
AuthenticationMethod=1
BulkLoadBatchSize=1024
Database=users
EnableBulkLoad=0
EnableQuotedIdentifiers=0
HostName=monserver.com
PortNumber=1433
```

Le paramètre *Driver* pointe sur l'emplacement du pilote ODBC SQL SERVER (`S0sqls26.so`)

Vous pouvez utiliser l'outil « *isql* » pour effectuer cette vérification de connexion avec la SGBD.

```
isql -v SQL_SERVER login pass
+-----+
```

```

| Connected! |
+-----+
SQL> select count(*) from class;
+-----+
| 19 |
+-----+
SQLRowCount returns 4294967295
1 rows fetched

```



Pour pouvoir valider votre connexion via ISQL, vous devez avoir préalablement défini les variables d'environnements LD\_LIBRARY\_PATH (ou LIBPATH sous AIX), ODBC\_HOME et ODBCINI pour le compte utilisateur exécutant ISQL.

### 3.8.2. Configuration SAS

Une fois que la connexion à la source ODBC, via iSql est opérationnelle, vous devez modifier le fichier sasenv\_local de SAS ( /SASFoundation/9.4/bin)

```

export LD_LIBRARY_PATH=/usr/local/lib:/usr/lib:/home/sas/SQLSERVER_ODBC_DRIVER/
export ODBC_HOME=/home/sas/ODBC/
export ODBCINI=/home/sas/ODBC/odbc.ini

```

Dans une session SAS, vous pouvez alors valider la bonne prise en compte de ces variables, en soumettant le code ci-dessous :

```

%put %quote(%sysget(LD_LIBRARY_PATH));
%put %quote(%sysget(ODBCINI));
%put %quote(%sysget(ODBC_HOME));

```

Il est également possible de positionner la valeur de sa variable LD\_LIBRARY\_PATH (ou LIBPATH sous AIX) en l'exportant dans sa session UNIX puis en vérifiant le « linkage » de ces binaires SAS :

```

ldd /sas/install/SASFoundation/9.4/sasexe/sasodb

```

Vous obtenez le résultat suivant :

```

linux-vdso.so.1 => (0x00007fff51313000)

```

```
libodbc.so => /usr/local/lib/libodbc.so (0x00007fa7c3944000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007fa7c3717000)
...
```

Si, au contraire, vous obtenez le résultat suivant, c'est que votre binaire SAS ne parvient pas à trouver sa variable LD\_LIBRARY\_PATH ou que celle-ci ne pointe pas sur un emplacement contenant la librairie dynamique libodbc.so :

```
linux-vdso.so.1 => (0x00007fff267bd000)
libodbc.so => not found
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f8ae94b3000)
```

### 3.9. Check-list des tâches à réaliser avant validation

Le tableau ci-dessous récapitule l'ensemble des vérifications à effectuer afin de valider sa connexion à ODBC.

La première étape consiste à vérifier que le pilote ODBC est bien installé sur la machine où SAS est installé, et valider la connexion sans utiliser SAS. Sous Windows, voir le chapitre 6.2 Configuration pas à pas sous Windows et, sous Unix, le chapitre 6.3 Configuration pas à pas sous Unix.

Vérifiez la licence des produits SAS/ACCESS to ODBC. Depuis une session SAS, utilisez le code ci-dessous pour vérifier si le produit est bien inclus dans la licence :

```
proc setinit; run;
```

En exécution, le journal doit lister les modules sous licence.

```
-----
SAS Access To Oracle SAS Access To ODBC
```

Vérifiez physiquement la présence des binaires du module SAS/ACCESS Interface to ODBC :

- Sous Windows, le fichier sasodb.dll doit être présent dans !SASROOT/access/sasexe.
- Sous Unix, le fichier sasodb doit être présent dans le répertoire !SASROOT/sasexe

Vérifier ou positionner les variables d'environnement.

### 3.10. Validation de la connexion à ODBC, dans une session SAS

Une fois les prérequis validés, vous pouvez vérifier la connexion avec la base depuis SAS. Pour cela, il suffit maintenant d'affecter une bibliothèque ODBC en utilisant une instruction « LIBNAME » avec un certain nombre de paramètres de connexion.

La syntaxe est la suivante :

```
LIBNAME libref odbc <options de connexion > <options libname>;
```

L'ensemble des options utilisables lors de la création d'une bibliothèque ODBC, notamment les options LIBNAME sont expliquées sur la page de l'aide [LIBNAME Statement Specifics for ODBC](#).

Exemple de connexion :

```
LIBNAME test ODBC DSN=SQL_SERVER user=dbitest password=dbigrp1;
```

Vous devez obtenir l'affichage suivant :

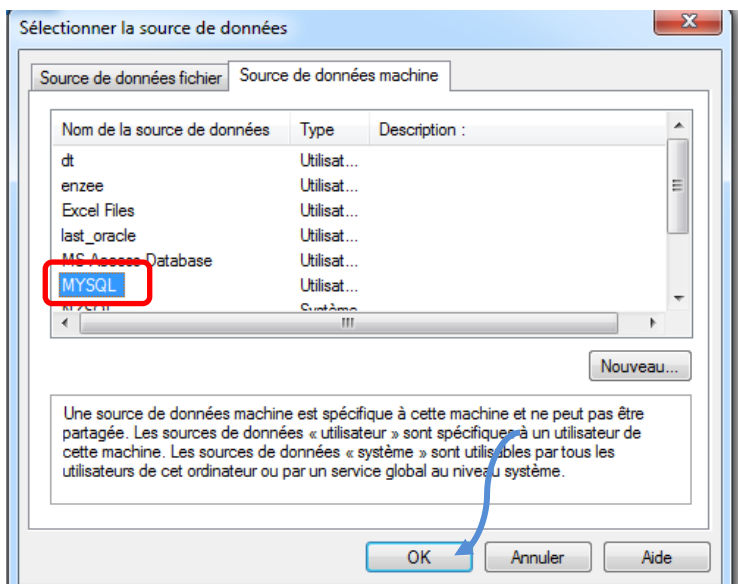
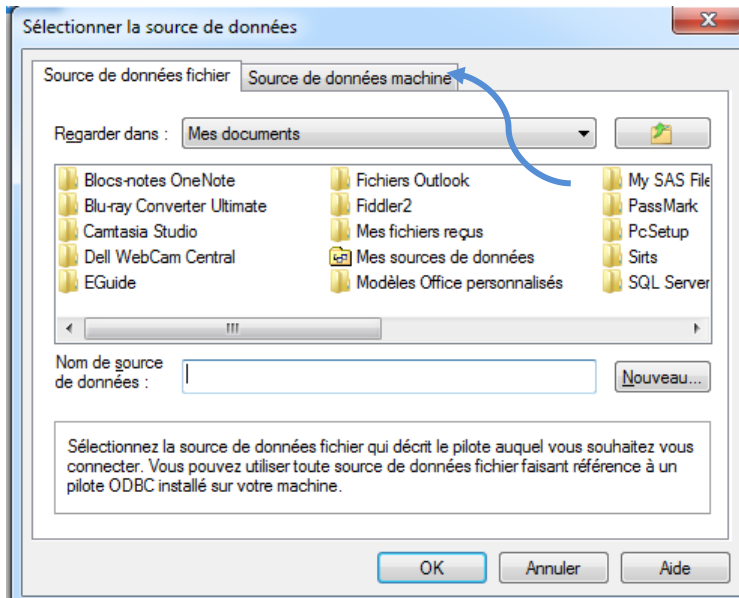
```
NOTE: Libref TEST was successfully assigned as follows:  
Engine:          ODBC  
Physical Name:  SQL_SERVER
```

Sur Windows, il existe une astuce permettant de retrouver les options de connexion. L'utilisation de l'option « prompt » ou « dbprompt » dans le LIBNAME ouvre l'interface interactive qui permet de saisir les valeurs pour les options appropriées par rapport au moteur que vous utilisez dans le LIBNAME. Attention toutefois, cette option fonctionne uniquement si SAS et la source ODBC sont définis localement.

La syntaxe est la suivante :

```
libname odbclib odbc prompt;
```

Vous pouvez ensuite manuellement choisir la source de données à laquelle vous souhaitez vous connecter :



Le journal indique la bonne création de votre bibliothèque ODBC :

```
NOTE: Libref ODBCLIB was successfully assigned as follows:  
Engine: ODBC  
Physical Name: MYSQL
```

Vous pouvez vérifier les contenus des variables macros « sysdbrc » qui contiennent le code (retourne 0 si tout c'est bien passé) et « sysdbmsg » qui retourne le message associé avec le code retourné en provenance de la base de données. Il existe aussi sqlxrc et sqlxmsg pour la méthode pass-through.

```
%put %superq(sysdbmsg);
```



Vous obtenez le résultat suivant :

```
ODBC: DSN=MYSQL;SERVER=localhost;UID=root;PWD=pass;DATABASE=test;PORT=3306
```

Et vous pouvez utiliser ce résultat comme paramètre à votre LIBNAME

```
Libname odbclib ODBC PROMPT="DSN=MYSQL;SERVER=localhost;UID=root;PWD=pass;DATABASE=test";
```



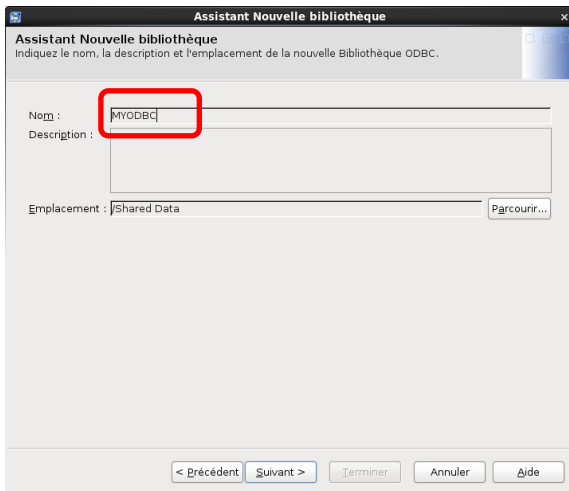
Les options COMPLETE, PROMPT et REQUIRED ne fonctionnent pas sous environnement UNIX mais UNIQUEMENT sous SAS Windows.

### 3.11. Création d'une bibliothèque ODBC dans les Métadonnées

The screenshot shows the SAS Management Console interface. In the left-hand tree view, the path is: SAS Management Console > Gestion de l'environnement > Gestionnaire de bibliothèques de données > Bibliothèques. The 'Nouvelle bibliothèque...' option is highlighted with a red box. Below this, the 'Assistant Nouvelle bibliothèque' dialog box is open, displaying a list of database types. 'Bibliothèque ODBC' is selected and highlighted with a red box. Other options in the list include 'Bibliothèque DB2 pour hôtes z/OS', 'Bibliothèque de données Aster', 'Bibliothèque Greenplum', 'Bibliothèque Hadoop via Hive', 'Bibliothèque Informix', 'Bibliothèque Microsoft Excel', 'Bibliothèque Microsoft SQL Server pour hôtes UNIX', 'Bibliothèque MySQL', 'Bibliothèque Netezza', 'Bibliothèque OLE DB', 'Bibliothèque Oracle', 'Bibliothèque Postgres', 'Bibliothèque SAP HANA', 'Bibliothèque Sybase', 'Bibliothèque Sybase IQ', 'Bibliothèque Teradata', 'Bibliothèque Vertica', 'Bibliothèque générique', 'Bibliothèque préaffectée', 'Bibliothèque SAS', 'Bibliothèque de compatibilité SAS V6', and 'Bibliothèque de moteur DISTANT SAS/SHARE'. At the bottom of the dialog, there are buttons for '< Précédent', 'Suivant >', 'Terminer', 'Annuler', and 'Aide'.

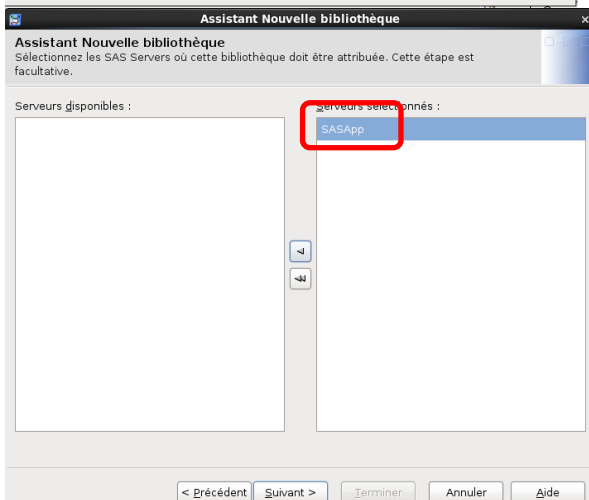
Une fois connecté dans la SAS Management Console, rendez-vous dans « Gestion de l'environnement » > « Gestionnaire de bibliothèques de données » > « Bibliothèques » et « Nouvelle bibliothèque »

Dans la première fenêtre de l'assistant bibliothèque, sélectionnez « Bibliothèque ODBC »

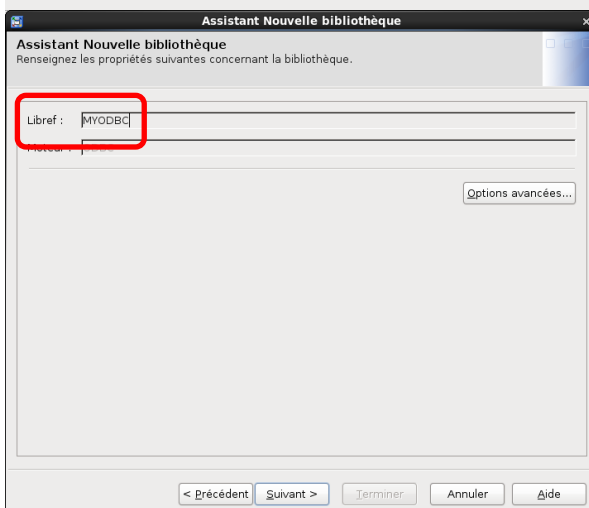


Indiquez le nom, la description et l'emplacement de la nouvelle bibliothèque ODBC.

Exemple : *MYODBC*

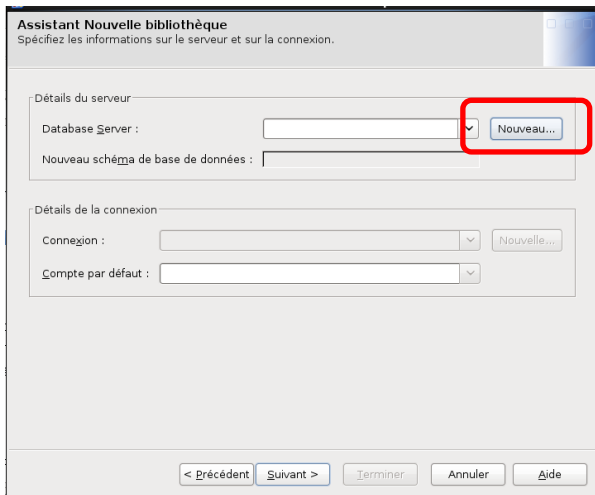


Sélectionnez les SAS Servers où votre nouvelle bibliothèque doit être attribuée.



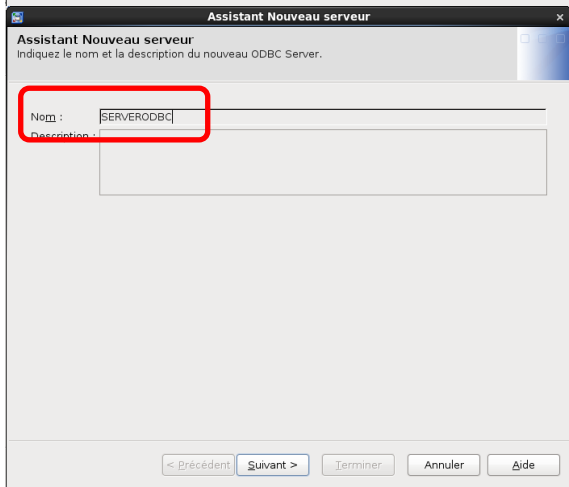
Renseignez le nom du « libref » concernant la bibliothèque.

Exemple : *MYODBC*



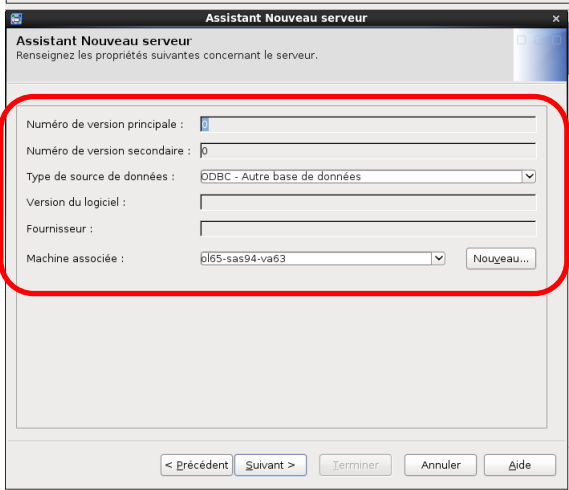
Nous devons maintenant saisir les informations sur le serveur et sur la connexion.

Si vous n'avez pas encore créé de serveur, cliquez sur le bouton « Nouveau »



Indiquez le nom et la description du nouveau serveur ODBC.

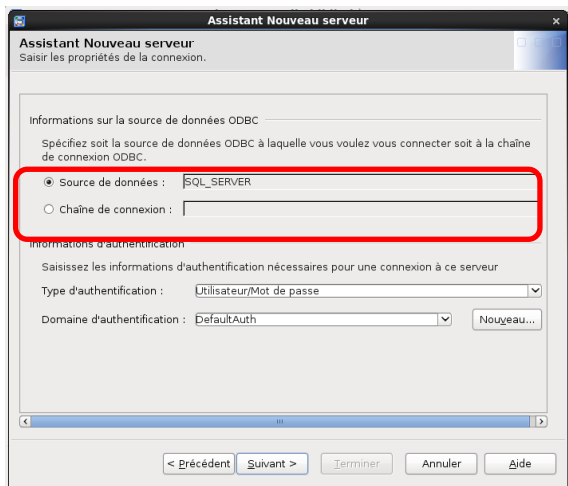
*Exemple : Serveur ODBC*



Renseignez les propriétés concernant le serveur :

- Numéros de versions
- Type de source de données
- Version du logiciel
- Fournisseur

Et choisissez la « Machine associée »

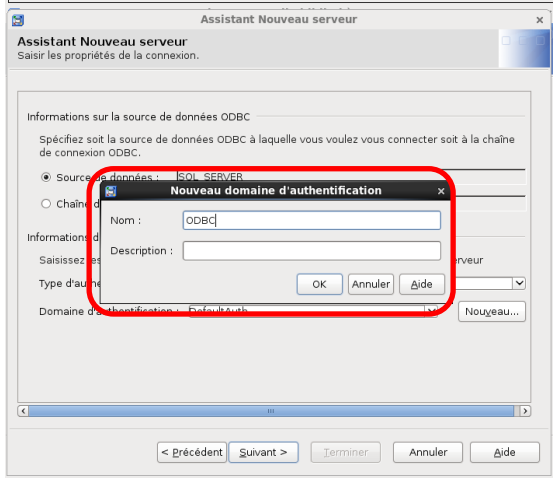


Nous allons maintenant saisir les propriétés de la connexion et plus précisément les informations sur la source de données ODBC.

Spécifiez :

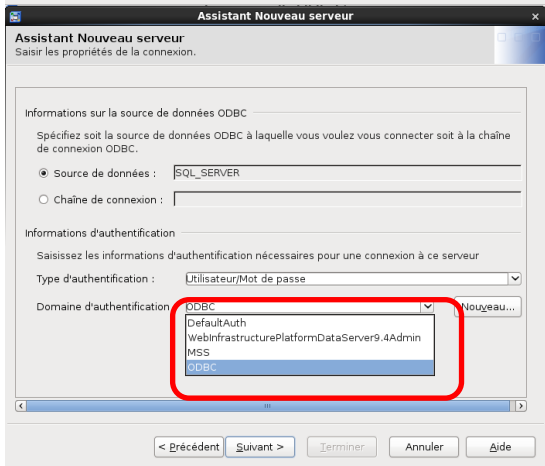
- Soit la source de données ODBC à laquelle vous voulez vous connecter,
- Soit la chaîne de connexion ODBC.

*Exemple : SQL\_SERVER (qui correspond à mon entrée DSN dans mon fichier ODBC.ini)*

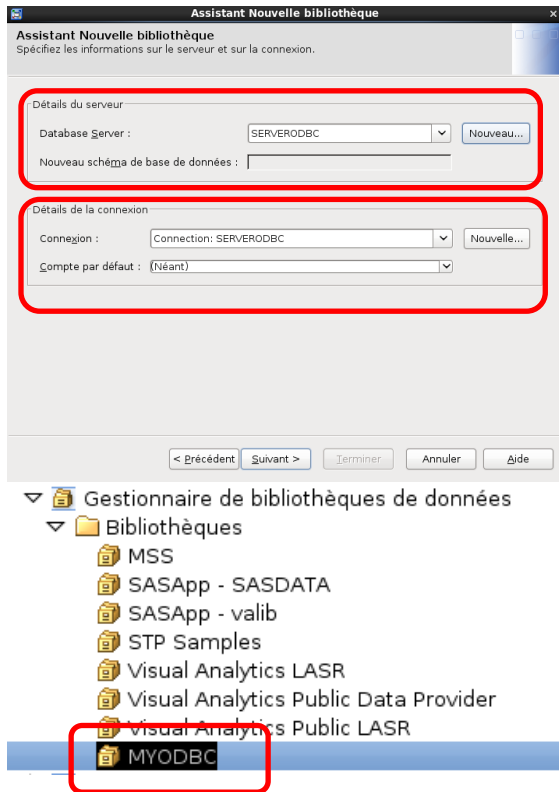


Créer ensuite un Nouveau domaine d'authentification pour pouvoir associer les comptes ODBC à vos utilisateurs SAS.

*Exemple : ODBC*



Une fois créé, sélectionnez le bon domaine.



Spécifiez les informations sur le serveur et sur la connexion.

*Exemple : Database Server : SERVERODBC*

*Connexion : SERVERODBC*

Félicitation votre bibliothèque ODBC est maintenant créée !

Maintenant que notre bibliothèque ODBC est créée dans les métadonnées, nous pouvons tester son accessibilité.

Lançons une session SAS et soumettons le code suivant :

```
options metaserver=localhost
      metaport=8561
      metaprotocol=bridge
      metauser=sasdemo
      metapass=sasdemo
      metarepository='Foundation' ;

libname MYODBC META library="MYODBC";
```

Nous obtenons la confirmation suivante :

```
NOTE: Libref MYODBC was successfully assigned as follows:
      Engine:          META
      Physical Name:   SQL_SERVER
```

### 3.12. Se connecter à une base sans configurer de DSN ODBC

Comme nous l'avons vu dans les chapitres précédents, lorsque vous utilisez l'interface SAS/ACCESS to ODBC, il n'est pas nécessaire de configurer une source de données (DSN) pour attribuer une bibliothèque à votre base de données (<http://support.sas.com/kb/52/777.html>). Par exemple, lorsque vous utilisez le pilote ODBC MySQL, vous pouvez établir une connexion sans DSN en utilisant la syntaxe suivante :

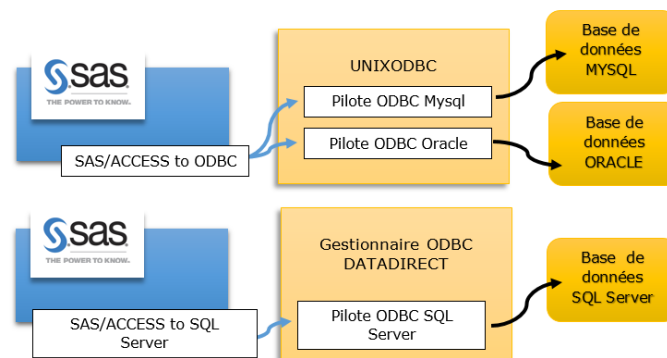
```
libname mydata odbc COMPLETE="driver=MySQL ODBC 5.3 ANSI Driver;
                               user=xxxx;
                               pwd=xxxx;
                               database=test;
                               Server=localhost" ;
```



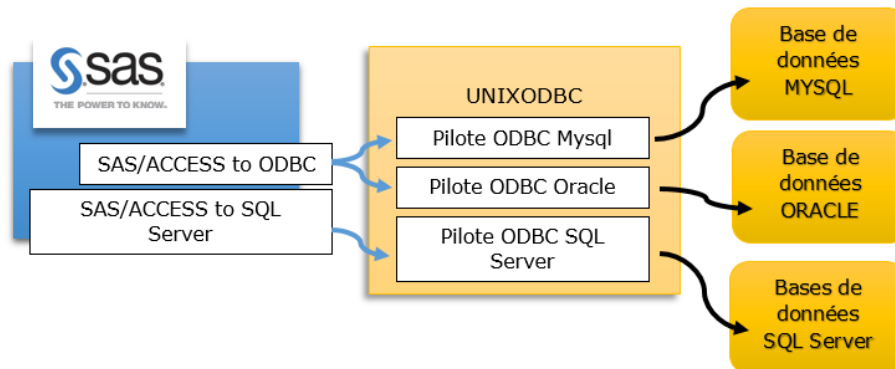
Cette option ne fonctionne que sous Windows.

### 3.13. Faire cohabiter SAS/ACCESS to SQL Server et SAS/ACCESS Interface to ODBC sous Unix

Lors de la mise en place de vos connexions, vous pourriez être confronté à cette problématique : comment, sous UNIX, mettre en place une connexion vers ODBC et une connexion vers SQL Server ? En effet, le module SAS/ACCESS Interface to SQL Server s'appuie sur un [package ODBC Datadirect fourni par SAS](#). Ce package comprend un gestionnaire ODBC et un pilote ODBC pour SQL Server. Ce gestionnaire ODBC fonctionne exclusivement avec les pilotes Datadirect. Il n'est donc pas possible d'y déclarer les informations de connexion vers une base s'appuyant sur un pilote qui n'est pas Datadirect, comme il n'est pas possible de faire pointer son environnement sur deux gestionnaires ODBC.



La solution pour faire cohabiter plusieurs connexions ODBC utilisant plusieurs fournisseurs de pilote ODBC est d'utiliser un gestionnaire ODBC autorisant l'utilisation de pilotes d'origine diverse, comme UNIXODBC :



Ainsi, votre fichier ODBC.ini contient les références à l'ensemble des vos sources ODBC.

Il est également possible d'utiliser uniquement SAS/ACCESS to ODBC. Dans ce cas, vous devez vous procurer votre propre pilote ODBC pour SQL Server et le faire fonctionner dans le gestionnaire ODBC. Dans ce cas plusieurs solutions sont possibles :

- DataDirect fournit les composants clients nécessaires ODBC pour les plates-formes UNIX/LINUX. Ils peuvent vous octroyer une licence pour un pilote ODBC pour MS SQL Server et ils fournissent également un gestionnaire de pilotes ODBC. Data Direct offre des pilotes ODBC pour diverses sources de données ODBC et se sont des pilotes fiables, faciles à mettre en œuvre et bien supportés.
- Easysoft et OpenLink sont également des fournisseurs de pilote ODBC proposant un gestionnaire de pilotes ODBC et un pilote ODBC pour MS SQL Server pour les plates-formes Unix/Linux.
- Microsoft fournit également aussi un pilote ODBC gratuit pour MS SQL Server pour Linux (uniquement sous Linux). Si vous décidez d'utiliser le pilote Microsoft ODBC sous Linux, vous aurez besoin d'UnixODBC. Toutefois, SAS supporte UnixODBC version 2.3.2 mais les pilotes Microsoft Sql Server 13 a besoin d'une version de UNIXODBC inférieur à 2.3.2 (inférieur à 2.3.1 pour la version 11). Ainsi, il vous faudra « forcer » l'installation du pilote Microsoft avec la version 2.3.2 mais gardez à l'esprit que les environnements ODBC freeware / shareware peuvent-être plus compliqué à configurer et qu'il n'y a pas de support officiel pour ces offres.

### **3.14. La boîte à outils SAS/ACCESS to ODBC**

Les sous-chapitres suivants présentent des astuces ou les connaissances à avoir lorsque vous souhaitez travailler efficacement avec SAS/ACCESS to ODBC.

### **3.14.1. Définition d'un schéma lors d'une connexion via SAS/ACCESS to ODBC**

Lorsque vous utilisez l'interface SAS/ACCESS Interface to ODBC pour vous connecter à une base de données, le système SAS cherche les tables qui appartiennent ou qui ont été créées par l'ID utilisateur utilisé dans la déclaration LIBNAME. Si SAS ne trouve pas de tables associées à cet ID utilisateur, le journal SAS indique que la bibliothèque a été assignée avec succès :

```
NOTE: Libref MYLIB was successfully assigned as follows:  
Engine:          ODBC
```

Cependant, aucune table n'apparaît dans la fenêtre SAS Explorer alors que vous avez des tables bien présentes dans votre base de données :



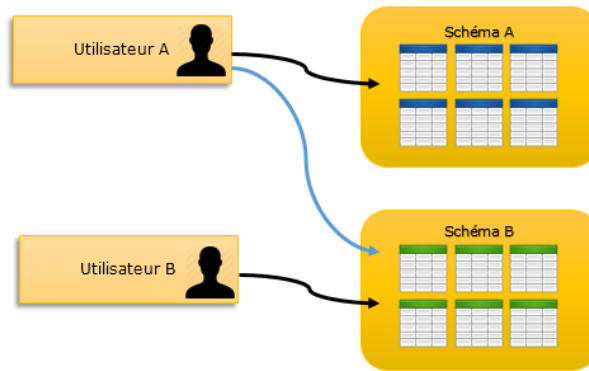
La bibliothèque ne montre pas de tables car, dans votre base de données, il n'y a pas de tables appartenant ou créées par l'utilisateur indiqué dans l'instruction LIBNAME. Par conséquent, vous devez définir un « schéma » dans votre instruction LIBNAME.

Le schéma est un ensemble d'objets d'une base de données, comme les tables, les vues, les fonctions ou tous autres objets définis. Un utilisateur est un compte qui a des droits sur ces objets. Un utilisateur peut avoir des droits sur tous les objets d'un schéma ou des droits restrictifs en lecture, exécution ou en modification, sur un ou plusieurs objets d'un schéma.

Mais un utilisateur peut ne pas avoir de schéma associé à son compte ou son schéma peut être une coquille vide. Cet utilisateur peut, si ses droits lui permettent, accéder aux objets d'autres schémas.

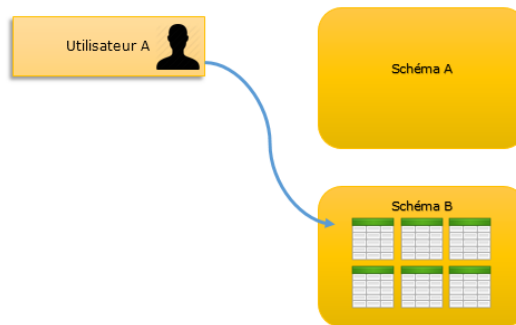
L'utilisateur A possède ses propres objets dans le schéma A mais il peut aussi accéder aux objets du schéma B, dont les objets sont gérés par l'utilisateur B :





L'utilisateur B a tous les droits sur ses objets (c'est le propriétaire du schéma et des objets créés) et donne des droits d'accès à l'utilisateur A.

Dans notre cas, l'utilisateur A n'a pas d'objet dans son schéma A :



Il souhaite voir les objets du schéma B. La syntaxe est la suivante :

```
libname mylib odbc user=utilisateurA password=password dsn=<MA BASE>  
schema=SchemaB;
```

Il existe également une méthode pour lister les schémas et objets accessibles par un utilisateur.

Soumettez le code suivant (par exemple avec Oracle) :

```
proc sql;  
  connect to odbc(user=scott password=tiger  
               dsn=oracle);  
  create table work.odbcSchema as  
  select * from connection to odbc  
  (ODBC::SQLTables);  
quit;
```

Vous obtenez une table, dans votre WORK, listant les objets accessibles et les schémas :

T	TABLE_SCHEM	TABLE_NAME	TABLE_TYPE
2856	SYS	XDS_ACL	SYSTEM VIEW
2857	SYS	XS_SESSION_ROLES	SYSTEM VIEW
2858	SYSTEM	PRODUCT_PRIVS	SYSTEM VIEW
2859	APEX_030200	WWW_FLOW_DUAL100	TABLE
2860	APEX_030200	WWW_FLOW_LOV_TEMP	TABLE
2861	APEX_030200	WWW_FLOW_TEMP_TABLE	TABLE
2862	CTXSYS	DR\$NUMBER_SEQUENCE	TABLE
2863	CTXSYS	DR\$OBJECT_ATTRIBUTE	TABLE
2864	CTXSYS	DR\$POLICY_TAB	TABLE
2865	CTXSYS	DR\$THS	TABLE
2866	CTXSYS	DR\$THS_PHRASE	TABLE
2867	DWH_COM	DIM_ABON	TABLE
2868	DWH_COM	DIM_ACTION	TABLE
2869	DWH_COM	DIM_CAISSE_LOCAL	TABLE
2870	DWH_COM	DIM_EDS_NIV1	TABLE
2871	DWH_COM	DIM_EDS_NIV2	TABLE
2872	DWH_COM	DIM_EDS_NIV3	TABLE
2873	DWH_COM	DIM_FAML_NIV1_PRDT	TABLE
2874	DWH_COM	DIM_FAML_NIV1_TYP_VM	TABLE



Le nom du schéma est CASE SENSITIVE.

### 3.14.2. Gestion des synonymes Oracle

La problématique des synonymes va de pair avec la problématique des schémas vue dans le chapitre précédent. Pour permettre l'identification des objets de la base de données, Oracle utilise les synonymes. Les synonymes permettent de pointer vers des tables, des vues, des procédures, des fonctions ou des packages. L'utilisateur a donc besoin de connaître uniquement le nom du synonyme vers lequel il pointe.

Il existe deux types de synonymes :

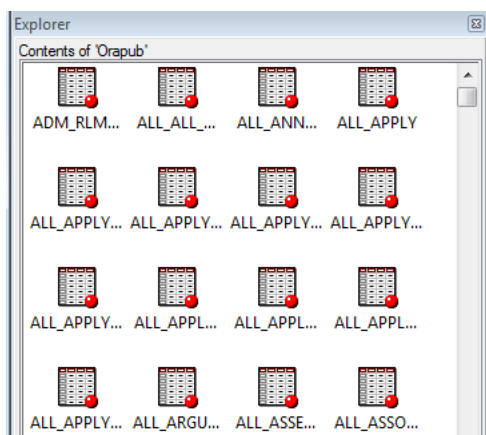
- Les synonymes publics, partagés par tous les utilisateurs
- Les synonymes privés, uniquement reconnus par un compte individuel.

Par défaut, les synonymes publics ne sont pas affichés par le moteur de LIBNAME avec l'interface SAS/ACCESS to ODBC.

Pour afficher et accéder aux synonymes publics, vous devez spécifier dans votre instruction LIBNAME les options PRESERVE\_TAB\_NAMES=YES et SCHEMA=PUBLIC. Notez qu'ils sont publics et ne sont pas associés à un schéma, bien qu'ils soient stockés sous « PUBLIC » dans les tables système Oracle :

```
libname ORAPUB odbc user=scott password=tiger dsn=oracle SCHEMA=PUBLIC  
PRESERVE_TAB_NAMES=YES;
```

Nous obtenons ainsi une bibliothèque permettant l'accès aux synonymes publics de la base Oracle. Cette bibliothèque liste l'ensemble des synonymes mais en fonction des droits de l'utilisateur, l'accès aux objets peut être limité :



Cette bibliothèque SAS ne contient que les synonymes. Pour avoir l'ensemble des objets, nous pouvons concaténer une bibliothèque (allocation classique n'affichant que les tables et les vues) avec cette bibliothèque créée ci-dessus.

La concaténation s'effectue avec l'instruction suivante :

```
libname ORAALL (ORAPUB ORA);
```

Le journal affiche le résultat suivant :

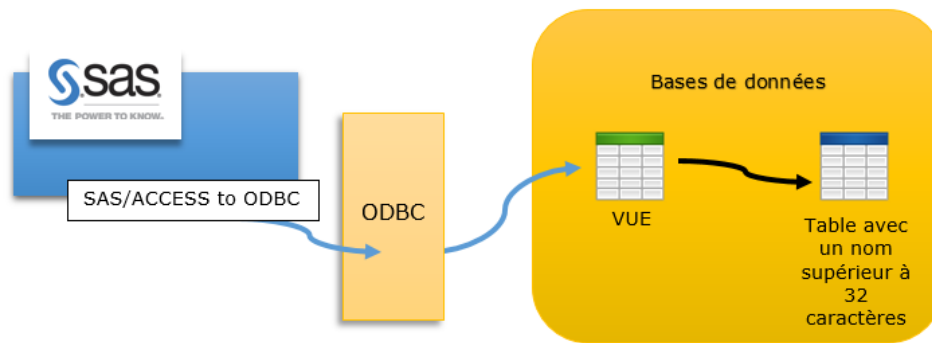
```
NOTE: Libref ORAALL was successfully assigned as follows:
Levels:                2
Engine(1):              ODBC
Physical Name(1):      oracle
Engine(2):              ODBC
Physical Name(2):      oracle
```

### ***3.14.3. Travailler avec des caractères spéciaux et des noms de plus de 32 caractères.***

Lorsque vous souhaitez travailler avec des tables de votre base de données, il peut arriver que certaines tables soient manquantes ou que le nom des colonnes de ses tables ne « reflète » pas ce qui est dans votre base de données. Plusieurs explications possibles :

- Le nom de votre table fait plus de 32 caractères (taille maximum pour un nom de table dans SAS)
- Le nom d'une ou plusieurs de vos colonnes de tables font 32 caractères ou contiennent des caractères spéciaux.

Si le nom de votre table fait plus de 32 caractères, il est possible d'y accéder en créant une vue, dans la base de données, pointant sur votre table de 32 caractères ou de passer par du [Explicit SQL Pass-Through](#).



Si le nom d'une ou plusieurs de vos colonnes de tables font 32 caractères ou contiennent des caractères spéciaux, vous pouvez utiliser l'option **VALIDVARNAME** et la positionner à **ANY**.

Imaginons que nous souhaitons accéder à une table MYSQL, dont la structure est la suivante :

#	Nom	Type
1	id	int(11)
2	champ_de_test_avec_une_longueur_superieur_a_32	int(11)
3	champ A \$	int(11)

Vérifions tout d'abord la valeur de l'option VALIDVARNAME en soumettant le code suivant :

```
proc options option=VALIDVARNAME;run;
```

Par défaut l'option est positionnée à **V7** :

```
VALIDVARNAME=V7    Specifies the rules for valid SAS variable names that can be
created and processed during a SAS session.
```

Si nous exécutons une **proc contents** sur notre table nous obtenons les informations suivantes :

Alphabetic List of Variables and Attributes						
#	Variable	Type	Len	Format	Informat	Label
3	champ_A__	Num	8	11.	11.	champ A \$
2	champ_de_test_avec_une_longueur_	Num	8	11.	11.	champ_de_test_avec_une_longueur_superieur_a_32
1	id	Num	8	11.	11.	id

Nous constatons que le nom de la variable ne « reflète » pas la réalité de la table. En effet, la colonne s'appelle « champ A \$ » et SAS a transformé les caractères spéciaux (les espaces et le \$) en « \_ »

Positionnons maintenant l'option **VALIDVARNAME** à **ANY**. Après avoir recréé notre bibliothèque, nous pouvons exécuter une **proc contents** :

Alphabetic List of Variables and Attributes						
#	Variable	Type	Len	Format	Informat	Label
3	champ A \$	Num	8	11.	11.	champ A \$
2	champ_de_test_avec_une_longueur_	Num	8	11.	11.	champ_de_test_avec_une_longueur_superieur_a_32
1	id	Num	8	11.	11.	id

Nous constatons maintenant que la variable porte le même nom que dans la base de données, à savoir « champ A \$ »

### 3.15. Quelle est la différence entre ODBC et OLE DB ?

ODBC a été conçu pour répondre aux besoins d'accès aux données de sources de type SQL. Ces sources de données sont généralement stockées dans des bases de données relationnelles.

ODBC a été élargi pour inclure des bases de données non-relationnelles qui ont leurs données stockées dans des colonnes et des lignes.

OLE DB est une technologie Microsoft, et également une API permettant l'accès à des données sous des formes diverses et variées – relationnelles, mais aussi non-relationnelles et multi-dimensionnelles – par l'intermédiaire d'une seule interface de programmation COM.

OLE DB était destiné à remplacer ODBC mais ce n'est pas le cas.

Plusieurs raisons :

- ODBC propose des pilotes très matures et fiables.
- OLE DB n'a pas ce niveau de maturité et ne bénéficie pas de la réputation rassurante d'ODBC.
- OLE DB ne fonctionne que sous environnement Windows.

### 3.16. SAS/ACCESS to ODBC et la technologie SAS In-database

La technologie [SAS In-Database](#) correspond à la capacité de SAS à pousser des traitements vers la base de données. Ainsi, lorsqu'une procédure (Proc FREQ ou proc SUMMARY par exemple) est soumise dans SAS avec en entrée une table du SGBD, SAS va « traduire » la procédure en langage SQL natif.

La technologie SAS In-Database n'est PAS supportée avec SAS/ACCESS interface to ODBC. Pour plus d'information, vous pouvez vous référer à la documentation SAS Introduction to SAS [In-Database Processing](#)

Pour plus d'informations sur la technologie SAS In-Database, vous pouvez lire l'article [SAS IN-DATABASE : QUAND LE TRAITEMENT SAS S'EXÉCUTE DANS LE SGBD](#)

### 3.17. Quelques problèmes connus et limitations avec SAS/ACCESS to ODBC

Cette section présente quelques erreurs typiques que vous pouvez rencontrer lors de la mise en place d'une LIBNAME ODBC et l'utilisation du moteur SAS/ACCESS interface to ODBC.

Pour chaque problème, nous fournissons les éléments à vérifier pour résoudre l'erreur.

#### 3.17.1. Problèmes lors de la création de la bibliothèque (via une instruction LIBNAME)

```
ERROR: The ODBC engine cannot be found.  
ERROR: Error in the LIBNAME statement.
```

- ⇒ Vérifier la présence du fichier sasiodb.dll dans *SASFoundation\9.4\access\sasexe* et procéder à l'installation si nécessaire.

```
ERROR: The SAS/ACCESS Interface to ODBC cannot be loaded. The SASODB code  
appendage could not be loaded.  
ERROR: Error in the LIBNAME statement.
```

- ⇒ Vérifier la présence du fichier sasodb dans *SASFoundation/9.4/sasexe/* et procéder à l'installation si nécessaire.

```
ERROR: Could not load /sas/install/SASFoundation/9.4/sasexe/sasodb (36 images  
loaded)  
ERROR: libodbc.so: cannot open shared object file: No such file or directory  
ERROR: The SAS/ACCESS Interface to ODBC cannot be loaded. The SASODB code  
appendage could not be loaded.  
ERROR: Error in the LIBNAME statement.
```

1. Vérifier les variables d'environnement (LD\_LIBRARY\_PATH, LIBPATH ou PATH en fonction de l'OS )
2. Vérifier que l'utilisateur exécutant la session SAS a les droits accès au répertoire contenant la librairie libodbc.so (ce répertoire est défini dans LD\_LIBRARY\_PATH)

```
ERROR: CLI error trying to establish connection: [unixODBC][Driver  
Manager]Data source name not found, and no default driver specified  
ERROR: Error in the LIBNAME statement.
```

Dans ce cas d'erreur, 4 points sont à vérifier :

1. Vérifier la présence de la variable d'environnement ODBCINI
2. Vérifier l'existence du fichier ODBCINI.ini référencé par ODBCINI
3. Vérifier que l'utilisateur exécutant la session SAS a les droits sur le fichier ODBCINI.ini référencé par la variable ODBCINI
4. Vérifier que la variable ODBCHOME est correctement définie et pointer sur le répertoire contenant le fichier ODBCINI.ini

```
ERROR: CLI error trying to establish connection: [unixODBC][SAS/ACCESS to SQL
Server][ODBC 20101 driver][20101]Login failed for user.
ERROR: Error in the LIBNAME statement.
```

- ⇒ La connexion à la base a fonctionné mais les informations de connexions fournies n'ont pas permis au module SAS/ACCESS d'accéder aux données proprement dites. Dans ce cas, vous devez vérifier, en dehors de SAS, si la connexion est possible avec le couple nom d'utilisateur/mot de passe utilisé dans SAS lors de la création du libname.

```
ERROR: CLI error trying to establish connection: [Microsoft][ODBC Driver
Manager] Driver's SQLAllocHandle on SQL_HANDLE_ENV failed
```

- ⇒ Ce message peut survenir lorsque vous essayez d'établir une connexion à une base de données DB2, en utilisant SAS/ACCESS to ODBC.
- ⇒ Pour contourner ce problème, vous devez ajouter un utilisateur DB2 dans le groupe DB2USERS, mettre à jour votre pilote DB2 jusqu'à la version 9.7 ou version ultérieure et installer une version complète du pilote DB2 (et non pas un client léger DB2). Vous trouverez plus de détails dans la note <http://support.sas.com/kb/49/615.html>

### **3.17.2. Problèmes pouvant survenir lors de la manipulation de données**

```
ERROR: CLI execute error: [Microsoft][SQL Server Native Client 11.0]Datetime
field overflow. Fractional second precision exceeds the scale specified in the
parameter binding.
```

- ⇒ Ce message d'erreur peut survenir lorsque vous utilisez une connexion ODBC à SQL Server l'option BULKLOAD=yes avec une table contenant un champ de type DATETIME2.
- ⇒ Ce problème n'apparaît pas si l'option BULKLOAD est positionnée à NO.
- ⇒ Ce problème apparaît avec SAS 9.4M1. Un hotfix pour SAS 9.4M1 est disponible à l'adresse suivante <http://ftp.sas.com/techsup/download/hotfix/HF2/N80.html#53865>



```
ERROR: An exception has been encountered.
Please contact technical support and provide them with the following traceback
information:
```

```
The SAS task name is [PRINT]
ERROR: Read Access Violation PRINT
Exception occurred at (02077902)
Task Traceback
Address      Frame      (DBGHELP API Version 4.0 rev 5)
0000000002077902 00000000246171B0 tkmk:tkBoot+0x15BC2
00000000020835BB 0000000024617210 tkmk:tkBoot+0x2187B
0000000002079BC8 0000000024617250 tkmk:tkBoot+0x17E88
000000000207770C 0000000024617290 tkmk:tkBoot+0x159CC
0000000002077921 00000000246172F0 tkmk:tkBoot+0x15BE1
0000000003445FB7 00000000246172F8 sashost:Main+0x1E497
```

- ⇒ SAS « crashe » lorsque vous utilisez une valeur supérieure à 100 pour les options de LIBNAME READBUFF ou INSERTBUFF  
(<http://support.sas.com/documentation/cdl/en/acreldb/67589/HTML/default/viewer.htm#p0u61lysy7qqzmn1bd66tz1om1yx.htm>)
- ⇒ Ce problème survient avec le pilote Oracle 11.2.x. Ce problème est identifié comme une anomalie dans le pilote Oracle.
- ⇒ Pour contourner ce problème, vous pouvez modifier la configuration du pilote en passant l'option « *Fetch Buffer Size* » de 64000 à 1024000.
- ⇒ Plus de détails dans NOTE 52761 : <http://support.sas.com/kb/52/761.html>

```
ERROR: Unable to load transient module (Bridge failure).
NOTE: The SAS System stopped processing this step because of errors.
```

Durant la mise en place de votre connexion à une base SQL Server, via ODBC, vous pouvez être confronté à l'erreur ci-dessus lorsque vous exécutez une PROC APPEND. Ce cas d'erreur ne se produit que dans certains cas et dans une combinaison d'options particulière dans la déclaration du libname. Aussi, ce comportement a été observé que lorsque toutes les conditions suivantes sont réunies:

- ⇒ La version de SAS est 32-bit.
- ⇒ Le type de données est inséré dans un type de données MONEY.
- ⇒ Le Bulkload est activé par la BCP = yes option.

Pour contourner ce problème, vous pouvez utiliser l'une des solutions suivantes :

- ⇒ Utilisez l'option INSERTBUFF à la place de l'option BCP.
- ⇒ Utilisez le type de données DECIMAL dans la table dans SQL Server au lieu du type de données MONEY.
- ⇒ Utilisez une version 64 bits de SAS.

```
ERROR: CLI describe error: [DataDirect][ODBC SQL Server Wire Protocol driver][Microsoft SQL Server]Statement(s) could not be prepared.
```

Ce message d'erreur, remonté par le driver ODBC à la session SAS, peut se produire lorsque vous essayez d'accéder à une table ayant un ou plusieurs espaces dans son nom. Pour accéder aux données de cette table et, afin qu'elle soit visible dans l'explorateur, essayez les options suivantes :

- ⇒ Ajouter l'option PRESERVE\_TAB\_NAMES positionné à YES et PRESERVE\_COL\_NAMES à YES
- ⇒ Si votre driver ODBC le permet, ajoutez l'option EnableQuotedIdentifiers à 1

#### **4. CONTACTER LE SUPPORT SAS EN CAS DE PROBLEME**

---

Si malgré toutes les informations contenues dans cet article, vous ne parvenez pas à faire fonctionner votre connexion via SAS/ACCESS to ODBC ou si vous rencontrez des erreurs (techniques, données mal formatées...) vous pouvez ouvrir un dossier au support SAS ([support@sas.com](mailto:support@sas.com)) en précisant :

- Votre version de SAS
- Le système d'exploitation
- Le gestionnaire ODBC utilisé
- Le pilote utilisé

Ajoutez à votre mail :

- Votre fichier Odbc.ini, si vous êtes sous Unix
- Une copie d'écran du gestionnaire ODBC, si vous êtes sous Windows
- Votre code.
- Le journal SAS contenant l'erreur
- Si possible des données pour reproduire, si le problème vous semble lié aux données.

#### **5. LIENS UTILES**

---

Quelques liens utiles pour aller plus loin:

[http://www.sas.com/offices/europe/france/services/support/faq/sasaccess\\_odbc.html](http://www.sas.com/offices/europe/france/services/support/faq/sasaccess_odbc.html)

<http://www2.sas.com/proceedings/forum2008/135-2008.pdf>

RECOMMENDED READING SAS/ACCESS 9.4 for Relational Databases: Reference, Third Edition.  
Available at <http://support.sas.com/documentation/cdl/en/acreldb/66787/PDF/default/acreldb.pdf>

Accessing a Microsoft SQL Server Database from SAS on Microsoft Windows  
<http://support.sas.com/kb/43/070.html>

Defining an ODBC Library in SAS® 9.2 Management Console Using Microsoft Windows NT Authentication : <https://support.sas.com/techsup/technote/ts802.pdf>

<https://support.sas.com/techsup/technote/ts765.pdf>

SQL Pass-Through Facility Specifics for ODBC :

<http://support.sas.com/documentation/cdl/en/acreldb/67589/HTML/default/viewer.htm#p1f29m86u65hken1deqcybowtgma.htm>

Le Système SAS et les accès via OLE DB : une introduction :

[http://www.sas.com/offices/europe/france/services/support/doc\\_techniques/introduction\\_systeme\\_sas\\_acces\\_ole\\_db.pdf](http://www.sas.com/offices/europe/france/services/support/doc_techniques/introduction_systeme_sas_acces_ole_db.pdf)

## **6. CONCLUSION**

---

J'espère que vous avez eu autant de plaisir à lire cet article que j'ai eu à l'écrire mais ce dont je ne doute pas, c'est que vous devez maintenant vous rendre compte des possibilités offertes par l'interface SAS/ACCESS to ODBC. Ainsi, comme vous l'avez lu, SAS/ACCESS Interface to ODBC est un produit très polyvalent qui vous permet d'accéder aux données provenant de nombreuses sources de données.

Ce document illustre à travers de nombreux exemples, comment se connecter à une source de données ODBC en utilisant des déclarations SAS « LIBNAME » et la procédure SQL.

La prochaine étape consiste à utiliser ces astuces et techniques pour explorer les sources de données ODBC que vous utilisez tous les jours.

Enfin, un dernier conseil : assurez-vous de lire la documentation SAS/ACCESS afin de profiter au maximum des possibilités offertes par SAS/ACCESS to ODBC.

**Nicolas HOUSSET**

Consultant Support Clients SAS France