



Gestion de l'accès aux données d'un serveur SPDS

The Power to Know.

Le stockage de données dans un serveur SPDS bénéficie de différents niveaux de sécurité :

1. Serveur : authentification au niveau du système d'exploitation.
2. Applicatif : authentification au niveau du Serveur SPDS
3. ACL : différents niveaux de sécurité au niveau des ressources SPDS
4. Audit : tracer les connexions/déconnexions des utilisateurs et leurs opérations.

1^{er} niveau :

Le lancement des processus SPDS doit se faire avec un compte dédié à l'administration du serveur SPDS.

L'ensemble des tables/vues créées auront les droits de ce compte.

2^{eme} niveau :

SPDS donne la possibilité de créer une base pour gérer les utilisateurs ; l'interface à utiliser est PSMGR (account manager).

Pour le lancer, il faut utiliser la syntaxe suivante :

```
psmgr !spdsroot/site
```

Cette interface permet donc:

- de déclarer les comptes SPDS et leurs mots de passes
- permet de limiter les connexions en fonction de l'adresse IP
- après trois essais de connexions , échec
- définir la date d'expiration du mot de passe
- fixer les niveaux d'autorisation : 0-7

3^{eme} niveau :

Définition des droits d'accès des utilisateurs SPDS sur les données du serveur.

Gestion des accès en fonction de différents profils :

- Group
- Utilisateur
- Table
- Colonne

Gestion des accès en fonction des droits :

- Lecture (READ)
- Ecriture (WRITE)
- Modification de la structure (ALTER)
- Modification des Autorisations d'accès (CONTROL)

Par défaut, le créateur d'une ressource a tous les droits d'accès sur cette ressource. Il peut également donner des droits d'accès à d'autres utilisateurs. Un utilisateur ayant un niveau d'habilitation suffisant, peut donner à d'autres utilisateurs des droits sur des ressources qui ne lui appartiennent pas (aclspecial=yes ou ayant droit de CONTROL sur les ressources).

Différents types d'acls peuvent être créés :

Générique : affecte un ensemble de ressource avec l'utilisation de wildcards : ex : vent*

Domaine : acls attribuées systématiquement sur un domaine

Persistante : par défaut, lorsqu'une ressource est détruite, les acls sont effacés ; avec cette option, il est possible de les conserver.

4^{eme} niveau :

Les actions utilisateurs peuvent être collectées dans les fichiers de log du serveur

Possibilité supplémentaire de déclencher une fonction d'audit.

Le fichier ainsi créé permet de constituer une base de connexion/déconnexion/actions utilisateurs.

RESSOURCES ACL

Les ressources ACL sont assignées en utilisant la **Proc SPDO** ; l'utilisation de la proc SPDO est possible seulement à travers le Système SAS.

Cette procédure s'utilise sur un libref connecté à SPDS :

```
libname test sasspds "travail" server=luynes.spds user="user1"  
prompt=yes;  
proc spdo lib=spds;
```

SET ACLTYPE – Le défaut c'est DATA. Si l'on souhaite modifier un autre type de ressource :

```
set acltype catalog
```

SET ACLUSER – L'utilisateur avec lequel on va affecter les ressources par défaut est celui utilisé dans la connexion (user1)

La commande **ADD** est utilisée pour ajouter une ACL :

Exemple 1 – Ajout d'une acl sur un domaine :

```
add acl/libname read groupwrite;
```

Cette acl donne les droits de lecture universelle et d'écriture pour les groupes

Exemple 2 – Ajout d'une acl sur une ressource :

```
add acl ventes/read write;
```

Cette acl donne les droits de lecture/écriture universelle sur la ressource ventes

Exemple 3 – Ajout d'une acl générique :

```
add acl mi/generic read;
```

Lecture sur toutes les ressources commençant par mi*

Exemple 4 – Ajout d'une acl sur une colonne :

```
add acl mine.salary/groupread;
```

Lecture pour le groupe et accès refusé pour les autres.

Exemple 5 – Ajout d'une acl sur un catalogue :

```
Set acltype catalog ;  
Add acl moncat/read groupread groupwrite ;  
Lecture universelle et lecture/écriture pour le groupe.
```

La commande **MODIFY** permet de modifier une acl existante :

Exemple 1 – Modification d'une acl sur un domaine :

```
modify acl/libname ralph=(y,y,n,n);  
Modification d'accès au domaine pour un utilisateur désigné.
```

Exemple 2 – Modification sur une table :

```
modify acl class/nowrite user1=(y,n,n,n) user2=(y,n,n,n);  
Refus d'écriture pour tout le monde, sauf deux utilisateurs.
```

Exemple 4 – Modification de toutes les acls :

```
modify acl _all_/user1=(y,,,);  
Lecture sur toutes les ressources pour un utilisateur.
```

La commande **LIST** affiche les acls.

Exemple 1 – Liste toutes les acls affectées

```
list acl _all_;
```

Exemple 2 – Liste pour une acl générique

```
list acl ven/generic;
```

Exemple 3 – Liste sur toutes les colonnes d'une table

```
list acl ventes._all_;
```

Exemple 4 – Liste de toutes les colonnes de toutes les tables

```
list acl _all_._all_;
```

La commande **Delete** est utilisée pour détruire les acls :

Exemple 1 – Destruction des acls sur un domaine :

```
delete acl/libname;
```

Exemple 2 – Destruction des acls pour toutes les ressources :

```
delete acl _all_;
```

Exemple 3 – Destruction des acls sur une table :

```
delete acl cars ;
```

Exemple 4 – Destruction d'une acl générique (pour tout objet commençant par ca:

```
delete acl ca/generic ;
```

Exemple 5 – Destruction de l'acl d'une variable :

```
delete acl cars.model ;
```

SPDS permet de gérer les sécurités d'accès sur les lignes

Le créateur de la table exécute une vue avec la clause de sélection qu'il veut attribuer aux autres utilisateurs. Donc, modification des acls sur la vue et non pas sur la table !

```
/* Déclaration du libname SPDS */  
  
Libname test sasspds 'travail' server=luynes.spds user='user1' passwd='xxx' ;  
  
/*Création de la vue */  
  
Proc sql;  
Connect to sasspds (dbq='travail' server=luynes.spds user='user1'  
passwd='xxx');  
execute(create view carsvue as select * from master where model=' espace') by  
sasspds;  
quit;  
  
/*Modification des acls */  
/*Droits en lecture pour le group */  
  
proc spdo lib=test;  
set acluser user1;  
set acltype view;  
add acl carsvue/groupread read;  
quit;
```

Exemple de macro permettant de donner les droits en lecture sur une vue au group support :

```

%macro INIT_ACL(vue=) ;
  /* On détruit d'abords toutes les Acl sur l'objet */
  proc spdo lib=test;
    set acluser user1;
    set acltype view;
    delete acl &vue ;
  quit ;

  /* On recrée l'ACL */
  proc spdo lib=test;
    set acluser user1;
    set acltype view;
    /* add obligatoire avant le modify car il
    n'existait aucune ACL */
    add acl &vue ;
    modify acl &vue /_ALL_=(n,n,n,n);
  quit;
%mend INIT_ACL;

%macro GIVE_READ(groupe=, vue=);
  proc spdo lib=test;
    set acluser user1;
    set acltype view;
    modify acl &vue /&groupe=(y,n,n,n);
  quit ;
%mend GIVE_READ;

```

Les macros doivent être exécutées (compilées) avant leur utilisation. La macro INIT_ACL initialise les droits sur un objet donné (recréation de l'ACL avec interdiction à tout le monde). La macro GIVE_READ permet de donner les droits en lecture à un group ou à un user pour un objet donné. Leur utilisation se fera comme suit :

```

%INIT_ACL(vue=carsvue) ;
GIVE_READ(groupe=support , vue=carsvue) ;

```



SAS France
Domaine de Grégy - BP 5
77166 Grégy-sur-Yerres
Tél. : 01 60 62 11 11
Fax : 01 60 62 11 99

SAS Europe, Middle East & Africa
P.O. Box 10 53 40
Neuenheimer Landstr. 28-30
D-69043 Heidelberg, Germany
Tel: +49 6221 4160, Fax: +49 6221 474850

SAS, le Système SAS® sont les marques déposées de SAS Institute Inc., Cary NC, USA.
Les autres noms de produits ou concepts sont des marques déposées des sociétés respectives.

www.sas.com/france