

*Comment ?
(je fais pour ...)
(Oracle)*

Ou réponses à des questions qu'on se pose parfois.

1. TABLE DE MATIERES

1.	TABLE DE MATIERES	2
2.	SUPPRIMER UN JOB LANCE AVEC DBMS_JOB	3
3.	PROFITER DU NOLOGGING	4
4.	COMMENT J'UTILISE EN PL/SQL UNE PROCEDURE EXTERNE ECRITE EN C ?	5
4.1.	Le code source de extern.c	5
4.2.	Le code source de credIL.cmd	5
4.3.	Le code de la préparation SQL	5
4.4.	Configuration du listener	6
4.5.	Test	6
5.	CALCULER LA LONGUEUR D'UN LONG RAW.	7

2. SUPPRIMER UN JOB LANCE AVEC DBMS_JOB

Utiliser `dbms_job.remove(xx)`, ou `xx` est l'identifiant du job, trouvé avec :

```
| select job, what, log_user from dba_jobs;
```

C'est le créateur du job qui doit le supprimer!

3. PROFITER DU NOLOGGING

Des redo logs il y en aura toujours, a moins que Oracle ne change l'architecture du noyau.

Une transaction est systématiquement fliquée dans les redos.

Ce qui n'est pas fliqué peut être un éventail de DETAILS d'actions sur plusieurs objets (par exemple, truncate ne génère pas (beaucoup) de redo.

unrecoverable (V7) ou nologging (v8+) sont des appellations différentes pour les mêmes phénomènes (a peu près) en fonction de la version d'Oracle.

Il est nécessaire également d'ajouter un hint dans les opérations (exemple:

```
| INSERT /*+ APPEND */ ...
```

Une solution plus sioux est de mettre le tablespace qui contient la table en nologging: alter tablespace TBS nologging; Il y aura une information minimale dans les redo concernant les objets stockés. Petite vérification: select TABLESPACE_NAME, LOGGING from dba_tablespaces; (Vérifier la version ;-))

Le mode archivelog permet d'archiver les redo logs. Ne pas générer des entrées redo n'a rien a voir avec le mode d'archivage

4. COMMENT J'UTILISE EN PL/SQL UNE PROCEDURE EXTERNE ECRITE EN C ?

Très souvent on me demande comment lancer une commande OS à partir d'une procédure stockée. Voici une méthode:

4.1. LE CODE SOURCE DE EXTERN.C

```
#include <string.h>
__declspec(dllexport) void execute_cmd( char* cmd )
{
    system(cmd);
}
```

4.2. LE CODE SOURCE DE CREDLL.CMD

```
cl -I. /LD -Zi extern.c /link msvcrt.lib /nod:libcmt /DLL
copy extern.dll \\sclavos\m$\cextlib\
pause
```

La copie du fichier le place dans un répertoire connu par Oracle. Quant au compilateur, un bon cl de Visual Studio est suffisant ;-)

4.3. LE CODE DE LA PREPARATION SQL

```
set echo on
connect sys/xxx@ulib as sysdba
CREATE OR REPLACE LIBRARY EXECUTE_CMDLIB IS 'M:\cextlib\extern.dll';
/
CREATE OR REPLACE PROCEDURE execute_cmd (
    p_cmd IN VARCHAR2)
AS EXTERNAL
LIBRARY execute_cmdlib
NAME "execute_cmd"
PARAMETERS (p_cmd STRING);
/
show errors
--
grant execute on sys.execute_cmd to public;
drop public synonym execute_cmd;
create public synonym execute_cmd for sys.execute_cmd;
```


5. CALCULER LA LONGUEUR D'UN LONG RAW.

Comment calculer la longueur d'une colonne long raw ?

Passer par un champ intermédiaire (dans une table créée spécialement dans ce but) :

```
set serveroutput on;
CREATE TABLE blob_tab(blob_col BLOB);

INSERT INTO blob_tab (to_lob(colonne_long_raw) FROM testraw);
commit;
```

(Le champ colonne_long_raw de la la table testraw est de type long raw).

```
begin
  for x in (select blob_col from blob_tab) loop
    dbms_output.put_line(dbms_lob.getlength(x.blob_col));
  end loop;
end;
/
```