

Rohmad.net

Catatan Harian Seorang Pecinta Hidup

Menjalankan OS Command atau Shell Script dari PL/SQL

Posted on July 24, 2008 by Rohmad

Menjalankan OS Command atau shell script dari SQLPlus sudah biasa kita lakukan. Biasanya kita menggunakan **!** atau **host** (catatan: di Windows hanya bisa pakai **host**, tidak bisa pakai **!**), contoh:

```
SQL> ! ls -la
SQL> ! /data1/oracle/Users/rohmad/test.sh
SQL> host ls -al
SQL> host /data1/oracle/Users/rohmad/test.sh
```

Namun ini hanya berlaku:

- OS command ini eksekusi di mana kita menjalankan SQL Plus.
- OS command yang dijalankan adalah OS command di mana kita menjalankan SQL Plus. Misalnya kita menjalankan SQLPlus di PC kita, maka OS command yang dijalankan adalah OS command yang ada di PC kita, bukan OS command di mesin/server database

Tantangan berikutnya:

- bagaimana menjalankan OS command (shell script) yang ada di mesin/server database sementara kita memanggilnya lewat SQLPlus yang ada di PC kita?
- Bagaimana caranya menjalankan OS command (shell script) dari PL/SQL atau prosedur?

Nah, artikel ini akan menjawab tantangan tersebut. Kita akan memanfaatkan DBMS_SCHEDULER, feature ini mulai dikenalkan sejak Database Oracle versi 10g. Untuk versi 9i ke bawah, bisa mencari referensi di asktom.oracle.com

* * *

Misalkan saya punya script:

```
/data1/oracle/Users/rohmad/test.sh
```

Content script ini adalah:

```
#!/usr/bin/ksh
cd /data1/oracle/Users/rohmad/
/usr/bin/echo "Hello `date` " >> test.log
```

Persiapan

Saya ingin menjalankan script tersebut melalui PL/SQL. Oleh Oracle, shell script tersebut dipandang sebagai external script (bukan scriptnya Oracle).

Khusus di Solaris dan Linux (saya tidak tahu untuk system Unix lainnya): Untuk bisa menjalankan external script, pastikan bahwa Oracle bisa menjalankan **external job**; berikut ini step-step untuk meng-enable-kan external job tersebut:

1. Lihat file \$ORACLE_HOME/rdbms/admin/externaljob.ora

```
cd $ORACLE_HOME/rdbms/admin
ls -la externaljob.ora

-rw-r----- 1 oracle dba externaljob.ora
```

Kemudian lihat isinya:

```
more externaljob.ora

run_user = nobody
run_group = nobody
```

2. Dengan user root, ubah owner dan mode dari file externaljob.ora tersebut

```
chown root externaljob.ora
chmod 640 externaljob.ora
```

Edit file externaljob.ora, ganti *run_user* dan *run_group* menjadi user dan group dari Oracle Installation. Dalam contoh ini, Oracle Installation adalah milik user *oracle* dengan group *dba*

```
run_user = oracle
run_group = dba
```

3. Lihat file \$ORACLE_HOME/bin/extjob

Dengan user root, ubah owner dan mode

```
cd $ORACLE_HOME/bin
chown root extjob
chmod 4750 extjob
```

Dalam contoh ini, shell script tersebut akan dijalankan oleh user TEST. Beri privilege ke user TEST

```
SQL> conn system
SQL> grant create job to test;
SQL> grant create EXTERNAL job to test;
```

Langkah Utamanya

Caranya mudah sekali. Jalankan dengan block PL/SQL berikut ini:

```
connect TEST

BEGIN
DBMS_SCHEDULER.CREATE_JOB (
job_name=>'testjob',
job_type=>'EXECUTABLE',
job_action=>'/data1/oracle/Users/rohmad/test.sh',
enabled=>true,
auto_drop=>true);
end;
/
```

Intinya adalah:

1. Buat job dengan DBMS_SCHEDULER
2. Begitu di-create, jalankan job tersebut.
Ditunjukkan oleh parameter *enabled=>true*
3. Setelah job dijalankan, drop (hapus) job tersebut
Ditunjukkan oleh parameter *auto_drop=>true*

Kalau cara di atas terasa kepanjangan, kita bisa membuat prosedur-nya. Misalnya prosedur itu kita beri nama **jalankan**

```
Create or replace procedure jalankan (cmd in varchar2) as
Begin
DBMS_SCHEDULER.CREATE_JOB (
job_name=>'testjob',
job_type=>'EXECUTABLE',
job_action=> cmd,
enabled=>true,
auto_drop=>true);
end;
/
```

Sekarang, tinggal kita panggil prosedur **jalankan** tersebut

1. Bisa melalui command SQL berikut

```
exec jalankan ('/data1/oracle/Users/rohmad/test.sh');
```

2. Ataupun memanggilnya lewat block PL/SQL

```
begin
jalankan ('/data1/oracle/Users/rohmad/test.sh');
end;
/
```

Catatan :

- Baik dipanggil dari SQLPlus yang ada di server database maupun di client (PC kita), prosedur **jalankan** ini tetap menjalankan shell script (OS command) yang ada di server database.
- Gunakan SQL command berikut untuk melihat, apakah job yang kita buat itu berhasil apa tidak

```
select log_id, log_date, job_name, status, error#,
additional_info from user_scheduler_job_run_details
where job_name like 'TEST%';
```

- Gunakan SQL command berikut untuk memastikan bahwa job yang kita buat itu telah di-drop begitu selesai dijalankan

```
select job_name, job_type, job_action from
user_SCHEDULER_JOBS;
```

- Penting: Untuk bisa dijalankan oleh DBMS_SCHEDULER, paling tidak script tersebut harus bisa dijalankan lewat crontab. Tambahkan shell definition pada script, contoh kalau menggunakan shell KSH:

```
#!/usr/bin/ksh
```

Kalau memanggil perintah Oracle misalnya sqlplus atau sqldr, tambahkan parameter ORACLE_HOME dan PATH (misalkan ORACLE_HOME ada di /dirOracle/10.2.3)

```
export ORACLE_HOME=/dirOracle/10.2.3
export PATH=$ORACLE_HOME/bin:$PATH
```

Referensi:

1. [Oracle® Database Administrator's Guide 10g Release 2 \(10.2\) — Using the Scheduler](#)
2. [Oracle® Database PL/SQL Packages and Types Reference 10g Release 2 \(10.2\) — DBMS_SCHEDULER](#)

This entry was posted in [SQL and PL/SQL](#) and tagged [Database](#), [oracle](#), [PL/SQL](#), [SQL](#). Bookmark the [permalink](#).

One Response to *Menjalankan OS Command atau Shell Script dari PL/SQL*



izzudin says:

July 31, 2008 at 6:09 am

Thanks atas sharingnya Pak..mengenai cara menjalankan shell script di database. Nanti saya coba praktekkan....:).

Regard's

Izzudin Hanafie

[Reply](#)

Rohmad.net

Proudly powered by WordPress.