Practice 5

Performing Incremental Backups

Practice Target

In this practice you will use RMAN to create incremental backups and incrementally updated backups.

Practice Overview

In high level, in this practice, you will perform the following tasks:

- Take incremental database backups, differential and cumulative
- Enable Block Change Tracking (BCT) in ORADB database
- Create database incrementally updated backup
- Automate RMAN backup jobs in Linux and Windows platforms

Assumptions

This practice assumes the following:

- `srv1` appliance is up and running and its database `ORADB` is running in `OPEN` state
- `winsrv2` appliance is up and running and its database `ORAWIN` is running in `OPEN` state

Note

I recommend taking a snapshot of the appliances before you start implementing the practice.

A. Taking Incremental Database Backup

In the following steps you will gain experience on creating incremental backups.

1. Open Putty and login to `srv1` as `oracle`.

2. Invoke RMAN and connect to the local database as target.

```
rman target "'/ as SYSBACKUP'"
```

3. Take level zero incremental backup of the database.

```
BACKUP INCREMENTAL LEVEL 0 DATABASE TAG 'DBLVL0';
```

4. Write down the period of time taken for making level zero incremental backup.

5. List backupsets of the database. Examine the output and see how you can tell that the listed backupset is a level zero incremental backup.

```
LIST BACKUP OF DATABASE;
```

6. Write down the size of level zero incremental backupset file.

7. Run Swingbench to produce DML activities on the database.

a. In the hosting machine, open a command-line window and change the current folder to `$SWINGHOME\winbin`

b. Start Swingbench by issuing the following command:

```
set PATH=D:\oracle\product\12.1.0\client_1\jdk\jre\bin;%PATH%
swingbench.bat
```

c. Click on the "Start Benchmark run" (green Play button on the top left corner) button.

d. Wait for three minutes.

e. Stop the Benchmark Run by clicking on "Stop" button.

f. Exit Swingbench and close the command-line window

8. In the Putty session, take level one incremental backup of the database.

```
BACKUP INCREMENTAL LEVEL 1 DATABASE TAG 'DBLVL1';
```

- Is this a cumulative or a differential incremental backup?

9. Write down the period of time taken for making level one incremental backup.

10. List backupsets of database. Compare the sizes of level zero and level one incremental backups.

```
LIST BACKUP OF DATABASE;
```

11. Write down the size of the level one incremental backupset file.

12. Compare between the periods and sizes of the level zero and level one incremental backups.

13. Take level one incremental backup of the database.

```
BACKUP INCREMENTAL LEVEL 1 DATABASE TAG 'DBLVL1';
```

14. List backupsets of database. Compare between the sizes of the last two level one incremental backups.

Observe that the last incremental backup is much less in its size than the size of the previous level one incremental backup.

```
LIST BACKUP OF DATABASE;
```

15. Take a cumulative level one incremental backup of the database.

```
BACKUP INCREMENTAL LEVEL 1 CUMULATIVE DATABASE TAG 'DBLVL1';
```

16. List backupsets of database. Compare between the last two taken level one incremental backups.

```
LIST BACKUP OF DATABASE;
```

B. Enabling Block Change Tracking

In the following steps you will enable the block change tracking feature in the database and study its influence on the incremental backups.

17. Determine if block change tracking is enabled

```
SELECT status, filename FROM V$BLOCK_CHANGE_TRACKING;
```

18. Enable block change tracking:

```
ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;

# verify:
SELECT status, filename FROM V$BLOCK_CHANGE_TRACKING;
```

19. Take level one incremental backup of the database.

```
BACKUP INCREMENTAL LEVEL 1 DATABASE TAG 'DBLVL1';
```

20. Run the following query to check if the BCT has been used for the incremental backup.

```
SELECT USED_CHANGE_TRACKING, FILE#, AVG(DATAFILE_BLOCKS), AVG(BLOCKS_READ)
FROM   V$BACKUP_DATAFILE
WHERE  INCREMENTAL_LEVEL > 0
GROUP  BY USED_CHANGE_TRACKING, FILE# ORDER BY 1;
```

21. Take level one incremental backup of the database.

```
BACKUP INCREMENTAL LEVEL 1 DATABASE TAG 'DBLVL1';
```

22. Check if the BCT has been used for the incremental backup.

```
SELECT USED_CHANGE_TRACKING, FILE#, AVG(DATAFILE_BLOCKS), AVG(BLOCKS_READ)
FROM   V$BACKUP_DATAFILE
WHERE  INCREMENTAL_LEVEL > 0
GROUP  BY USED_CHANGE_TRACKING, FILE# ORDER BY 1;
```

Clean up

23. Delete all the backupsets.

```
DELETE BACKUPSET;
```

C. Creating Database Incrementally Updated Backup

In the following steps, you will create incrementally updated database backup.

24. Execute the following run block:

```
Run
{
 RECOVER COPY OF DATABASE WITH TAG 'incr_update';
 BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY WITH TAG 'incr_update' DATABASE;
}
```

25. Examine the output of the executing the run block in the previous step and try figuring out what RMAN has performed.

Observe that the output of executing the code above is a copy of the database data files.

26. Issue the following command to list the image copies of the database. Take a copy of the command output.

Observe that the only way to distinguish between the image copies that are generated by the `BACKUP AS COPY` command and the image copies that are generated by the "`BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY`" is the `TAG` value.

```
LIST COPY OF DATABASE;
```

27. (optional) Issue the following query.

The view `V$DATAFILE_COPY` can inform about incremental level of image files.

```
SELECT FILE#, STATUS, CREATION_TIME, COMPLETION_TIME,
TAG, BLOCKS*BLOCK_SIZE/1024/1024 SIZE_MB, INCREMENTAL_LEVEL
FROM V$DATAFILE_COPY
WHERE DELETED ='NO'
ORDER BY FILE#, RECID;
```

28. Execute the following command twice, just to generate some redo archive logs.

```
ALTER SYSTEM SWITCH LOGFILE;
```

29. Execute the run block again.

30. Examine its output again and see what RMAN has done.

Observe the following message from the run block execution output. This indicates that RMAN has created incremental level 1 backup.

```
channel ORA_DISK_1: starting incremental level 1 datafile backup set
```

31. List the backup sets of the database. Observe their type.

```
LIST BACKUPSET OF DATABASE;
```

32. Execute the following command twice, just to generate some redo archive logs.

```
ALTER SYSTEM SWITCH LOGFILE;
```

33. Execute the run block again (this is the third time).

34. Examine its output again and see what RMAN has done.

Observe that RMAN has applied the incremental backup taken previously on the image copies then it generated another incremental level 1 backup.

35. Issue the following command and take a copy of the command output. Compare it with the output of the same command taken earlier.

You will notice that the "Ckp SCN" has been incremented in the datafile copies, which is an indication that the datafiles copies have been updated.

```
LIST COPY OF DATABASE;
```

D. Creating an Automatic Backup Job in Linux

Regardless of the backup strategy that you implement in any environment, it ends up with scheduling a backup job to run frequently.

In the following steps you will create a backup job that runs every 4 hours starting from midnight.

Note

The target of this practice section is to demonstrate how to automate a backup job in Linux. The example in this practice section is taking an incrementally updated database backup. The contents of a backup job in a production system depend on the backup strategy used in that system.

36. As `oracle` user, create a directory to save the script files in it.

```
cd
mkdir scripts
```

37. Create a script that calls RMAN and executes the backup script, as follows:

```
vi ~/scripts/rman_script1.sh
```

```
#!/bin/bash
ORACLE_SID=ORADB; export ORACLE_SID
ORACLE_HOME=/u01/app/oracle/product/12.2.0/db_1; export ORACLE_HOME

$ORACLE_HOME/bin/rman log=/home/oracle/scripts/rman1.log append <<EOF
connect target '/ AS SYSBACKUP';
set echo on;
run {
 RECOVER COPY OF DATABASE WITH TAG 'incr_update';
 BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY WITH TAG 'incr_update' DATABASE;
}
Exit;
EOF
```

38. Set the execution permission on the file.

```
chmod 774 /home/oracle/scripts/rman_script1.sh
```

39. Open the `oracle` account `crontab` file and add the line that follows to it.

This setting executes the backup script every 4 hours starting from midnight.

```
# execute the following command:
crontab -e
```

```
# add the following line to it:
* 0,4,8,12,16,20 * * *  "/home/oracle/scripts/rman_script1.sh" > /dev/null
```

40. Execute the script to make sure it is working fine.

```
/home/oracle/scripts/rman_script1.sh
```

41. Check out the contents of the log file. If there is any error reported in the log file, fix it and perform the previous step again.

In a production system, you need to check the contents of the log file every time a backup is made. Consider automating sending it to your email.

```
vi /home/oracle/scripts/rman1.log
```

Clean up

42. Disable the backup job.

```
# execute the following command:
crontab -e
```

```
# add a remark symbol at the beginning of the following line:
#* 0,4,8,12,16,20 * * *  "/home/oracle/scripts/rman_script1.sh" > /dev/null
```

43. Delete the database backup images.

```
rman target "'/ as SYSBACKUP'"
DELETE COPY OF DATABASE;
```

44. You can shut down `srv1` at this stage. It is not needed anymore for this practice. If you took a snapshot of it at the beginning of this practice, consider deleting it.

E. Creating an Automatic Backup Job in Windows Platform

In the following steps you will create a backup job that runs every midnight in Windows environment.

Note

The target of this practice section is to demonstrate how to automate a backup job in Windows platform. The contents of a backup job in a production system depend on your backup strategy.

45. Login as `oracle` user to `winsrv2`.

46. Open a command-line window, invoke SQL*Plus and login to the database as sysdba.

```
sqlplus / as sysdba
```

47. Retrieve the FRA destination.

```
show parameter DB_RECOVERY_FILE_DEST
```

48. Open the file browser and change its current folder to the following folder.

`<fra_directory>\ORAWIN\ORAWIN`

49. Create a new folder in it and name it as `scripts`

50. Create a file in the new folder and name it as `rman.bat`

You may need to change the settings of the File Explorer to show file extensions.

51. Open the file with notepad (right-click on the file -> edit) and add the following code to it:

```
set ORACLE_SID=ORAWIN
set ORACLE_HOME=D:\oracle\product\12.2.0\dbhome_1

%ORACLE_HOME%\bin\rman cmdfile=D:\oracle\fra\ORAWIN\ORAWIN\scripts\rman.ora log=
D:\oracle\fra\ORAWIN\ORAWIN\scripts\rman.log append
```

52. Create a file in the new folder and name it as `rman.ora`

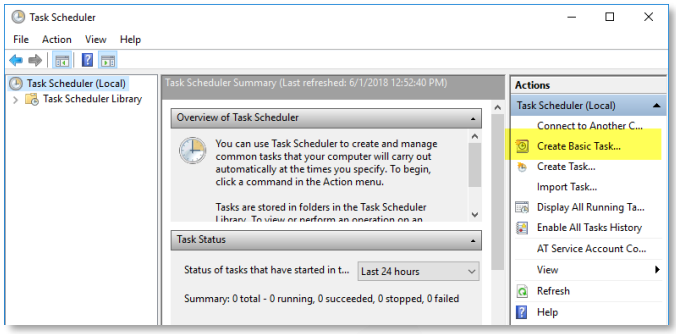
53. Open the file with notepad and add the following code to it:

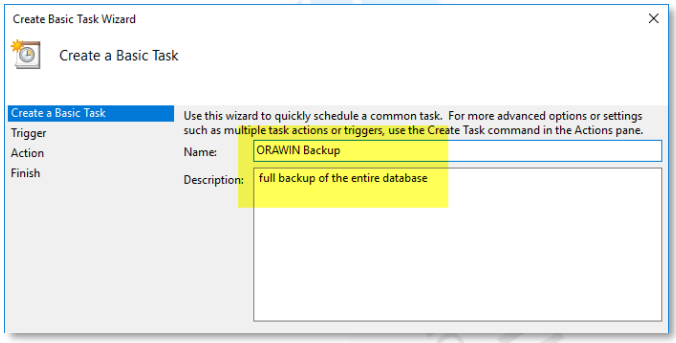
```
connect target '/ AS SYSBACKUP';
SET ECHO ON;

run {
  BACKUP DATABASE TAG 'FULL_DB';
}

Exit;
```

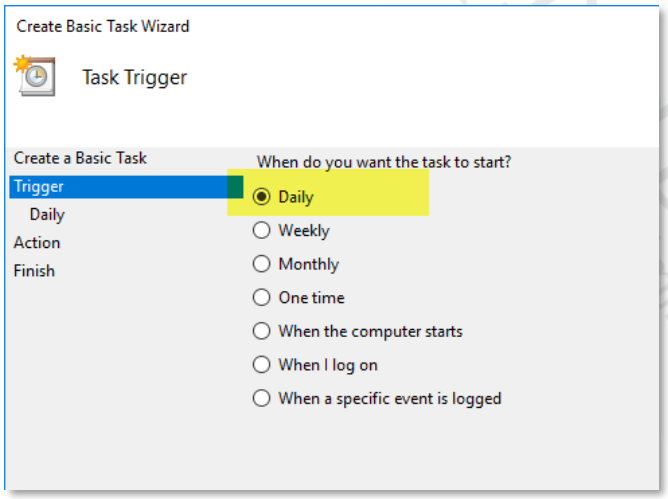
54. Open "**Task Scheduler**".

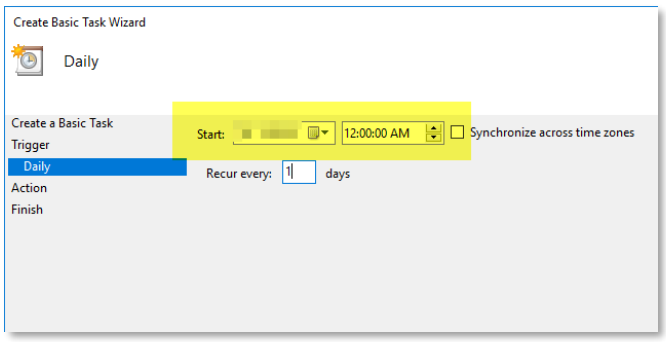
In `winsrv2`, press on [Windows] button on the keyboard then type "task". Windows should display the "Task Scheduler" shortcut for you.

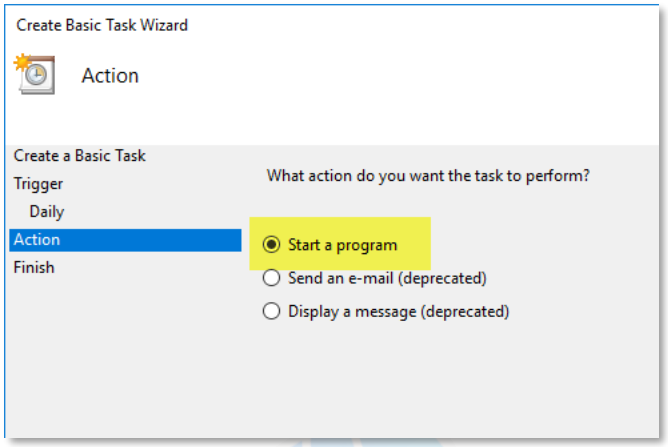
55. In "Task Scheduler", perform the steps as demonstrated in the following screenshots:

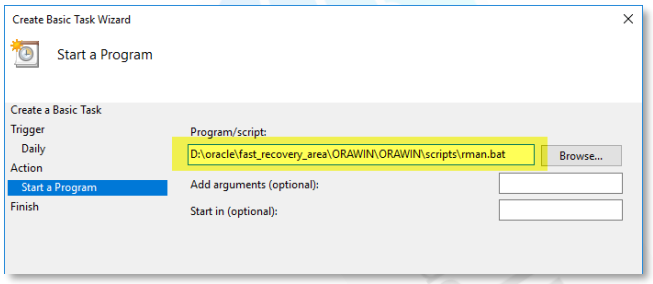


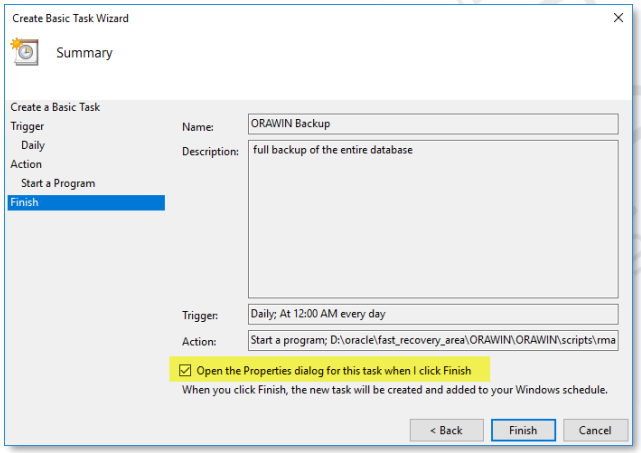


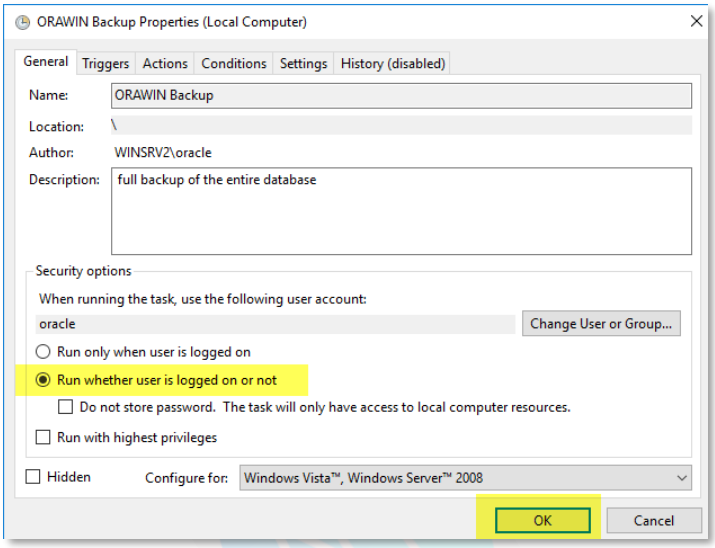


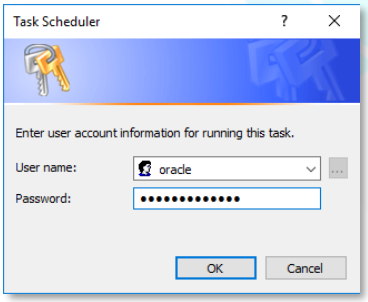


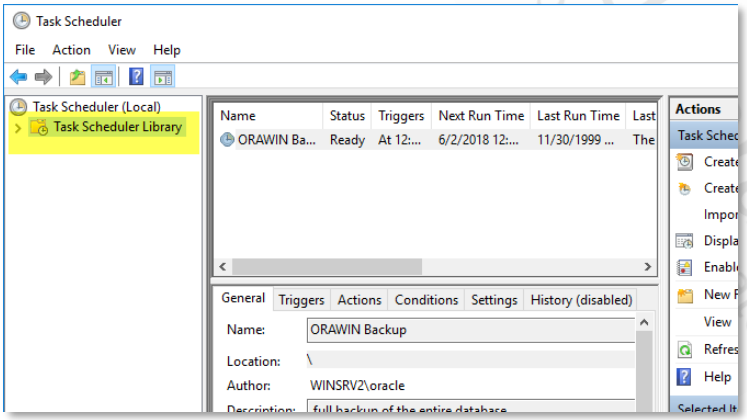

Create Basic Task Wizard
Action
Create a Basic Task
Trigger
Daily
Action
Finish
What action do you want the task to perform?
Start a program
Send an e-mail (deprecated)
Display a message (deprecated)


Create Basic Task Wizard
Start a Program
Create a Basic Task
Trigger
Daily
Action
Start a Program
Finish
Program/script:
D:\oracle\fast_recovery_area\ORAWIN\ORAWIN\scripts\rman.bat
Browse...
Add arguments (optional):
Start in (optional):

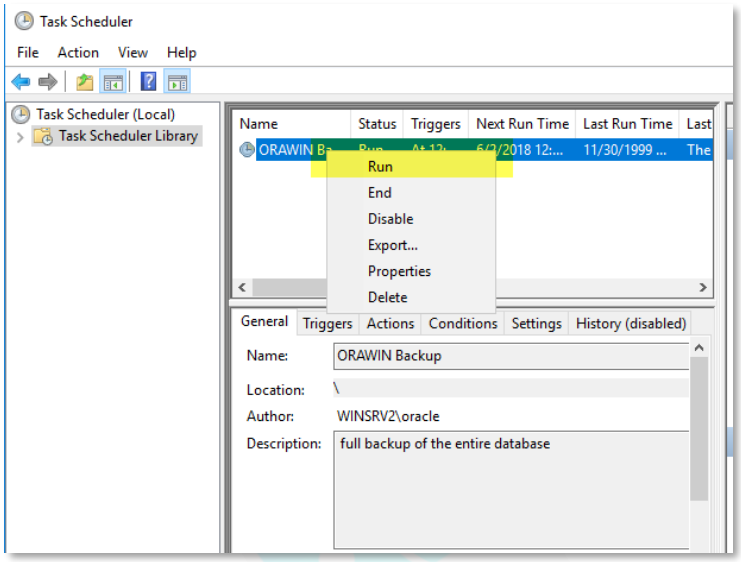

Create Basic Task Wizard
Summary
Create a Basic Task
Trigger
Daily
Action
Start a Program
Finish
Name:
ORAWIN Backup
Description:
full backup of the entire database
Trigger:
Daily; At 12:00 AM every day
Action:
Start a program; D:\oracle\fast_recovery_area\ORAWIN\ORAWIN\scripts\rma
Open the Properties dialog for this task when I click Finish
When you click Finish, the new task will be created and added to your Windows schedule.
< Back
Finish
Cancel


ORAWIN Backup Properties (Local Computer)
General
Triggers
Actions
Conditions
Settings
History (disabled)
Name:
ORAWIN Backup
Location:
\
Author:
WINSRV2\oracle
Description:
full backup of the entire database
Security options
When running the task, use the following user account:
oracle
Change User or Group...
Run only when user is logged on
Run whether user is logged on or not
Do not store password. The task will only have access to local computer resources.
Run with highest privileges
Hidden
Configure for:
Windows Vista™, Windows Server™ 2008
OK
Cancel


Task Scheduler
Enter user account information for running this task.
User name:
oracle
Password:
OK
Cancel


Task Scheduler
File
Action
View
Help
Task Scheduler (Local)
Task Scheduler Library
Name
Status
Triggers
Next Run Time
Last Run Time
ORAWIN Ba...
Ready
At 12:...
6/2/2018 12:...
11/30/1999 ...
Actions
General
Triggers
Actions
Conditions
Settings
History (disabled)
Name:
ORAWIN Backup
Location:
\
Author:
WINSRV2\oracle
Help

56. Test the task. Right-click on it and select "**Run**" from the display drop-down list.



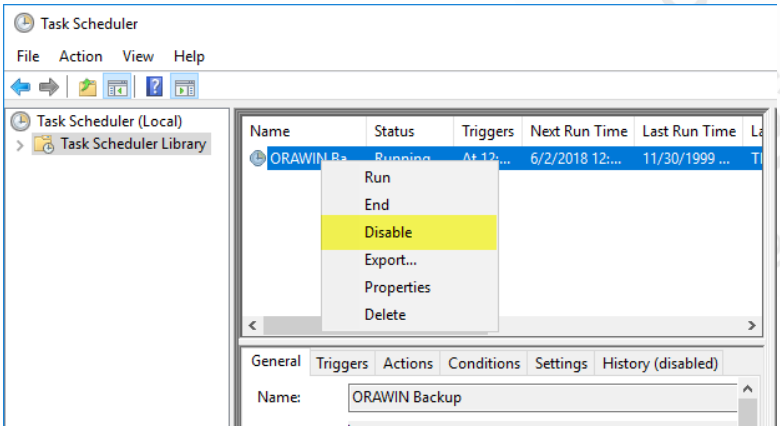
57. Check out the FRA folder. You should see the following sub-folders created by RMAN in it:

`BACKUPSET`
`AUTOBACKUP`

58. Check out the contents of `rman.log` file. It should contain output of the job execution.

Clean up

59. Disable the backup task. In Task Scheduler, right-click on the job and select "**Disable**".



60. Delete the database produced backup files.

```
cd \
rman target "'/ as SYSBACKUP'"
DELETE BACKUPSET;
```

Summary

In this practice, you gained practical experience on performing the following tasks:

- Take incremental database backups, differential and cumulative
- Enable Block Change Tracking (BCT) in ORADB database
- Create database incrementally updated backup
- Automate RMAN backup jobs in Linux and Windows platforms