DBA

RMAN

RMAN BACKUP SCHEDULED

Abstract: **We are implementing following backup strategies**

1. Weekly full backup (Backup full database once a week).
2. Daily incremental backup (Daily backup changes).

1. Introduction:

Performing regular, consistent backup of critical database is a important part of a database backup strategy. Oracle recovery manager is a client that can be used to perform backup and recovery tasks on your database. RMAN backups can be classified in following ways:

A. Full or Incremental

- Full Backup : A backup of datafile that includes every allocated block in the file being backed up.
- Incremental Backup : An incremental backup is either a level 0 backup, which includes every block in the file except blocks compressed out because they have never been used, or a level 1 backup, which includes only those blocks that have been changed since the parent backup was taken.

B. Open or Closed

- Open Backup : A backup of online, read/write datafiles when the database is open.
- Closed Backup : A backup of any part of the target database when it is mounted but not open. Closed backups can be consistent or inconsistent.

C. Consistent or inconsistent

- Consistent Backup : A backup taken when the database is mounted (but not open) after a normal shutdown. None of the datafiles has changes beyond its checkpoint. Consistent backups can be restored without recovery.

- Inconsistent Backup : A backup of any part of the target database when it is open or when a crash occurred or SHUTDOWN ABORT was run prior to mounting.An inconsistent backup requires recovery to become consistent.

Instead of performing the backup of database manually, we can automate the database backup process.We can individually backup data files, control files and redo log files.

We have scheduled backup by:

I. Creating batch file (.bat)that has RMAN backup command.
II. Creating a task which runs our batch file . (This can be done either using dbms_scheduler or using Windows Task Scheduler)

2. Creating a Batch File

2.1. Connect RMAN to the target database:

The database which we need to backup is connected using TARGET option.

RMAN> CONNECT TARGET SAMAN/SAMAN@SAMAN

Note: TARGET option takes user/password@service_id

Note: Also, RMAN only connects user with sysdba privilege.

2.2. Run block of commands:

RMAN allows to execute block of commands sequentially using the *run { block_of_commands; }* command.

```
RMAN> RUN {
        BLOCK_OF_COMMANDS;
      }
```

2.3. Allocate channel:

ALLOCATE CHANNEL manually allocates a channel, which is a connection between RMAN and a database instance. The ALLOCATE CHANNEL commands must be

issued within a RUN block because it allocates a channel only in the block where the command is issued.

We can allocate 255 channels where each channel can read up to 64 files in parallel.

Syntax of ALLOCATE CHANNEL is like:

ALLOCATE CHANNEL *channel_id* DEVICE TYPE DISK;

CHANNEL *channel_id* specifies a connection between RMAN and the target database instance. *Channel_id* is case-sensitive name of the channel. The database uses the *channel_id* to report I/O Errors.

DEVICE TYPE specifies the type of the storage for a backup. In our task, we have used DEVICE TYPE as DISK. When backing up to disk, we can spread the backup across several drives.

2.4. Set Limits:

We specifies the channel limits to any backup or copy using SET LIMIT CHANNEL commands.

Some Parameters of limit channel are:

- **Kbytes:** It specifies the maximum size in kilobytes of the backup pieces created on the channel. In our case, we have used 2097150 KB which is 1.999GB.
- **Maxopenfiles:** It controls the maximum number of input files that a backup command can have open at any given time. We use this parameter to prevent “Too many open files” operating system error messages when backing up a large number of files into a single backup set. Default number is 32 i.e. 32 input files can be opened concurrently. We also have used its value as 32.
- **Readrate:** It specifies the maximum number of buffers (each of size DB_BLOCKSIZE*DB_FILE_DIRECT_IO_COUNT) per second for backup or copy operation from each of the input datafiles. We use this parameter to “throttle back” RMAN (i.e set an upper limit for block reads) so that RMAN does does

not consume excessive disk bandwidth and thereby decline online performance. In our case, we have used its value as 200 that means, it can read 200 buffers per second from each input datafiles.

2.5. Set Maxcorrupt for datafiles:

It sets a limit on the number of previously undetected block corruptions that the database will permit in a specified datafile or group of datafiles. The default limit is zero, meaning that RMAN tolerates no corrupt blocks.

The `SET MAXCORRUPT` command specifies the total number of physical and logical corruptions permitted in a datafile during a backup job. If the sum of physical and logical corruptions detected for a datafile is no more than its `MAXCORRUPT` setting, then the `BACKUP` command completes. If more than `MAXCORRUPT` corrupt blocks exist, then RMAN terminates without creating output files.

Physical corruption (media corrupt) : Physical corruption can be caused by defective memory boards, controllers or broken sectors on a hard disk.

Logical corruption (soft corrupt) : Logical corruption can among other reasons be caused by an attempt to recover through a NOLOGGING action.

Syntax: SET MAXCORRUPT FOR DATAFILE <Data_File_Spec> TO <integer>

In our case, we have used

set maxcorrupt for datafile 1,2,3,4,5,6 to 0;

This means that datafile **1,2,3,4,5,6** will not tolerate more than **0** corruption. If, incase, any physical or logical corruption occur than the backup will fail and will not create any output file.

Note: If you specify `CHECK LOGICAL`, then the `MAXCORRUPT` limit applies to the sum of logical and physical corruptions detected. Otherwise, `MAXCORRUPT` only applies to the number of physical block corruptions.

2.6. Backup Database:

According to our plan, we have to perform incremental level 0(full backup) on sunday 6:00 AM and incremental level 1 backup on rest of the days in a week at 7:00 AM. We will iterate this on remaining weeks.

We have two files **sunday_fullbackup.bat** and **monday_2_saturday_incremental_backup.bat** which are triggered by **automate_sunday_backup.bat** file and **automate_mon_to_sat_backup.bat** file respectively.

sunday_fullbackup.bat

```
connect target saman/saman@saman;

run{
allocate channel d1 type disk;
setlimit channel d1 kbytes 2097150 maxopenfiles 32 readrate 200;
set maxcorrupt for datafile 1,2,3,4,5,6 to 0;
backup
incremental level 0
skip inaccessible
tag sunday_BACKUP
format 'C:\Oracle\flash_recovery_area\AUTOBACKUP\df_%Y_%M_%D_s%s_p%p'
database;
copy current controlfile to 'C:\Oracle\flash_recovery_area\AUTOBACKUP\cf_%Y_%M_%D_s%s_p%p.ctl';
sql 'alter system archive log current';
backup
format 'C:\Oracle\flash_recovery_area\AUTOBACKUP\al_%Y_%M_%D_s%s_p%p'
archivelog all
delete input;
release channel d1;
}
```

FIg 2.6.1: sunday_fullbackup.bat

monday_2_saturday_incremental_backup.bat

```
connect target saman/saman

run {
allocate channel d1 type disk;
setlimit channel d1 kbytes 2097150 maxopenfiles 32 readrate 200;
set maxcorrupt for datafile 1,2,3,4,5,6 to 0;
backup
incremental level 1
skip inaccessible
format 'C:\Oracle\flash_recovery_area\AUTOBACKUP\df_%Y_%M_%D_s%s_p%p'
database;
copy current controlfile to 'C:\Oracle\flash_recovery_area\AUTOBACKUP\cf_%Y_%M_%D_s%s_p%p.ctl';
sql 'alter system archive log current';
backup
format 'C:\Oracle\flash_recovery_area\AUTOBACKUP\al_%Y_%M_%D_s%s_p%p'
archivelog all
delete input;
release channel d1;
}
```

Fig 2.6.2. Monday_2_saturday_incremental_backup.bat

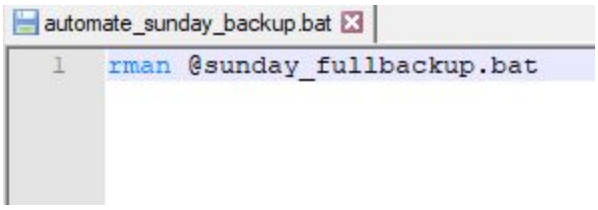


Fig 2.6.3. Automate_sunday_backup.bat

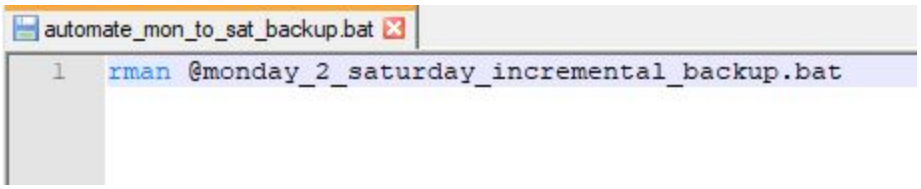


Fig 2.6.4. automate_mon_to_sat_backup.bat

In **sunday_fullbackup.bat** file, we have performed incremental level 0 (full backup). And in **monday_2_saturday_incremental_backup.bat** file, we have performed incremental level 1 backup.

Note: Incremental Backup ,by default, is differential, that means only the changes made from one day to another day are backed up thereby saving memory and time for backing up as well as recovery.

SKIP INACCESSIBLE:

By default, the `BACKUP` command terminates when it cannot access a data file. But we can specify parameters to prevent form termination.

SKIP INACCESSIBLE specifies RMAN to skip the datafiles that RMAN cannot read.

TAG: We must use tags to identify the datafile copies and incremental backups in our strategy so that they do not interfere with other backup strategies.

FORMAT: We use format clause to specify a filename format or an Automatic Storage Management disk group for a backup piece or image copy.

Some of the RMAN substitution variables are:

Syntax Element	Description
%Y	Specifies the year in this format: `YYYY`
%M	Specifies the month in the Gregorian calendar in format `MM`.
%D	Specifies the current day of the month from the Gregorian calendar in format `DD`.
%s	Specifies the backup set number. This number is a counter in the control file that is incremented for each backup set. The counter value starts at `1`and is unique for the lifetime of the control file. If you restore a backup control file, then duplicate values can result. Also, `CREATE CONTROLFILE`initializes the counter back to `1`.
%p	Specifies the piece number within the backup set. This value starts at `1` for each backup set and is incremented by 1 as each backup piece is created. **Note:** If you specify `PROXY`, then the `%p` variable must be included in the `FORMAT`string either explicitly or implicitly within %U.
%U	Specifies a system-generated unique filename (default). The meaning of `%U` is different for image copies and backup pieces. For a backup piece, `%U` specifies a convenient shorthand for `%u_%p_%c` that guarantees uniqueness in generated backup filenames.

COPY: Copy commands creates an image copy of a file. In our case, we have used it to copy the control file to the prescribed location.

2.7. SQL COMMAND:

We can use sql commands in RMAN from SQL command followed by a query inside the quotes(' '). However, we cannot use SELECT query (our environment: in Oracle 11g). We can use ALTER Statement, CREATE Statements.

We have used a query

ALTER SYSTEM ARCHIVE LOG CURRENT;

This command waits until the online redo log has completed the writing of the redo log file to the filesystem. This command is safer because it waits for the OS to acknowledge (ACK) that the redo log has been successfully written.

2.8. Backup Archive Logs:

We can use archive log backups to recover database transactions that have been lost due to an operating system or disk failure. By default, full backups include both data and archive logs, but we can also perform separate archive log backups.

We can use a unique backup piece format to avoid conflicting backup piece names by using FORMAT command.

2.9. Delete command:

It is not necessary to keep archived redo log files after backup. We can easily delete them, but they may be deleted by RMAN automatically after backup. To do this, we use the delete input command. Using this command, RMAN deletes the archived redo log files from only one destination.

If the delete all input command is used, all archived redo log files from all destinations will be deleted.

2.10. Release Channel:

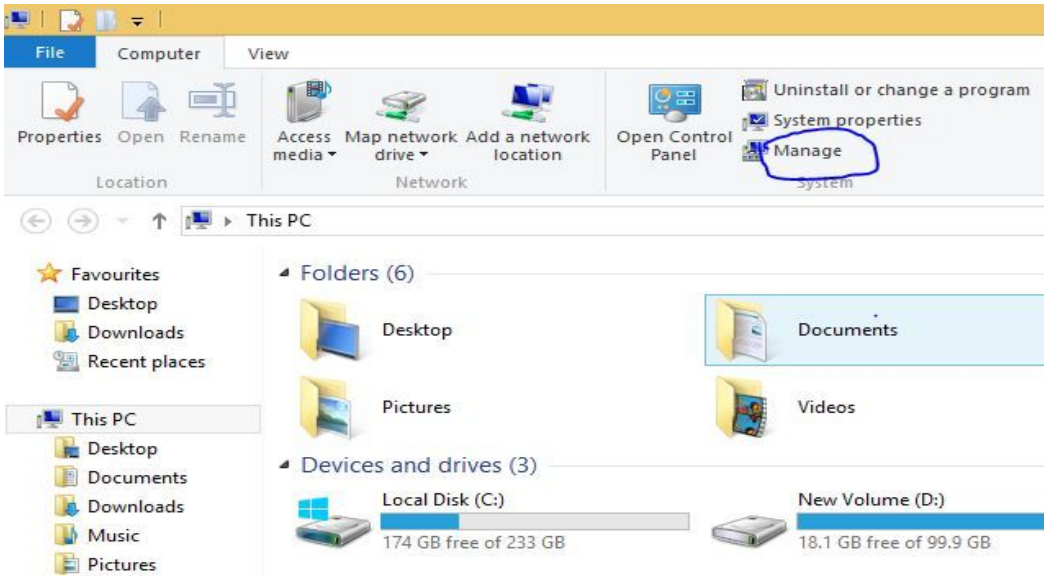
`RELEASE CHANNEL` command to release a normal or maintenance channel while maintaining a connection to a target database instance.

Syntax: **RELEASE CHANNEL *channel_id;***

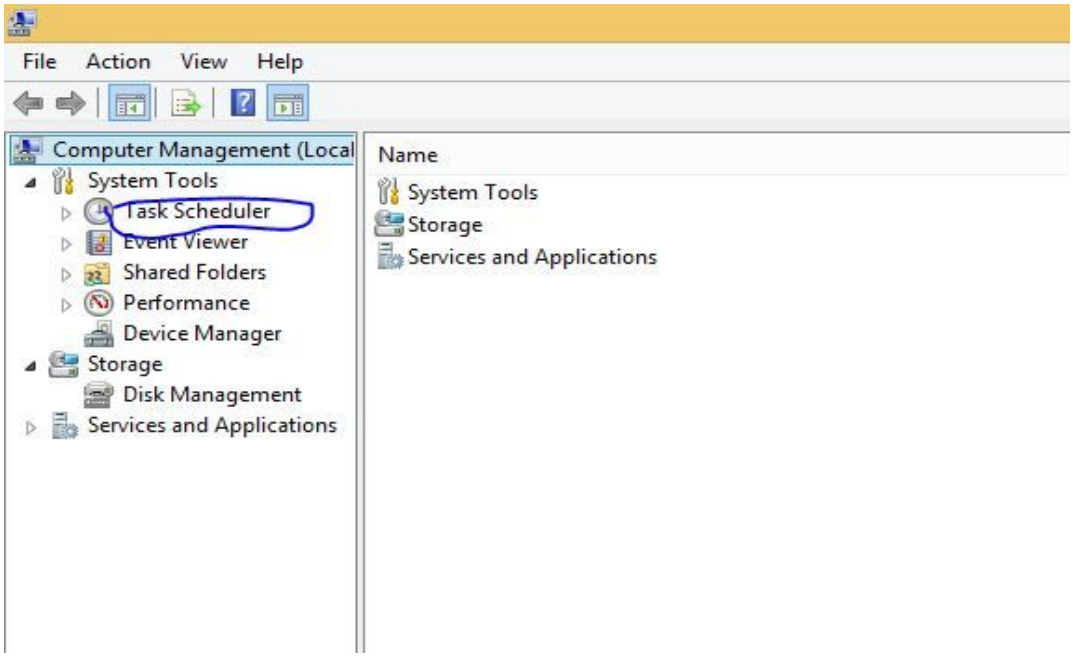
We have released the *channel_id* that was allocated at the beginning.

3. Scheduling jobs in windows scheduler

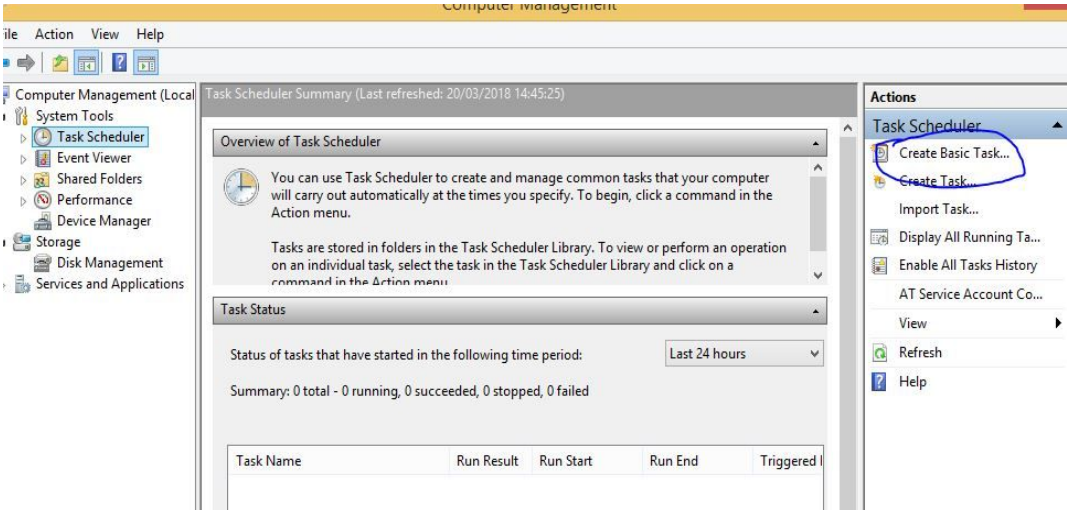
3.1. Open My Computer and click *manage*



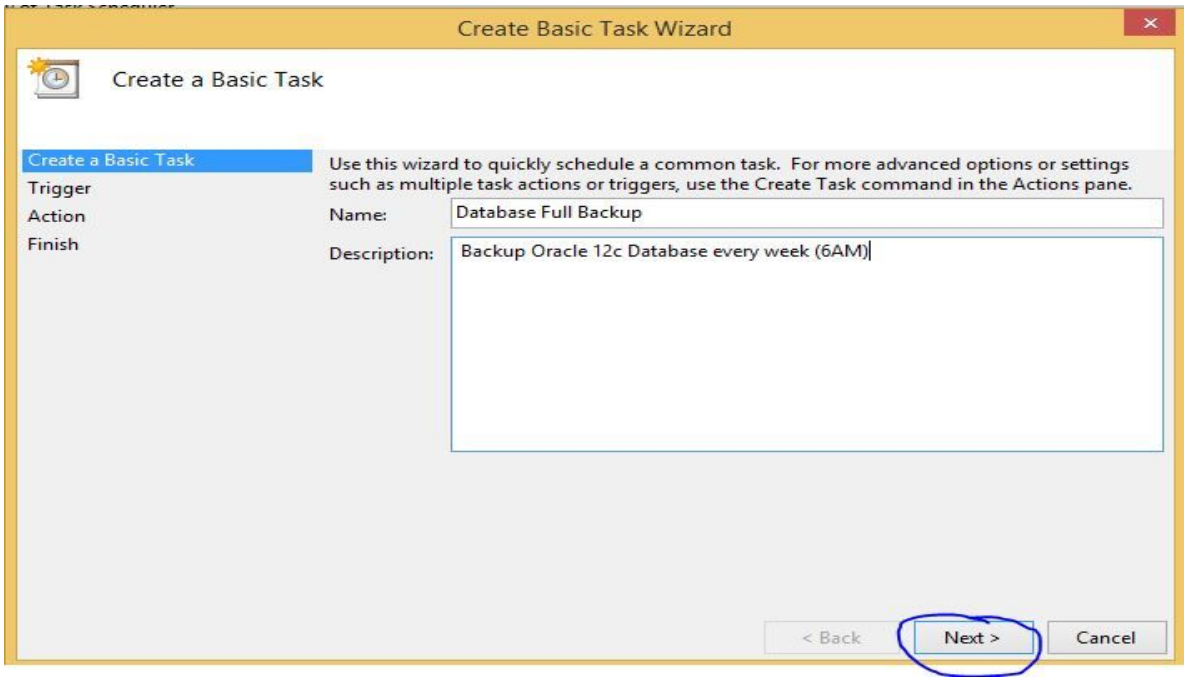
3.2. Click Task Scheduler



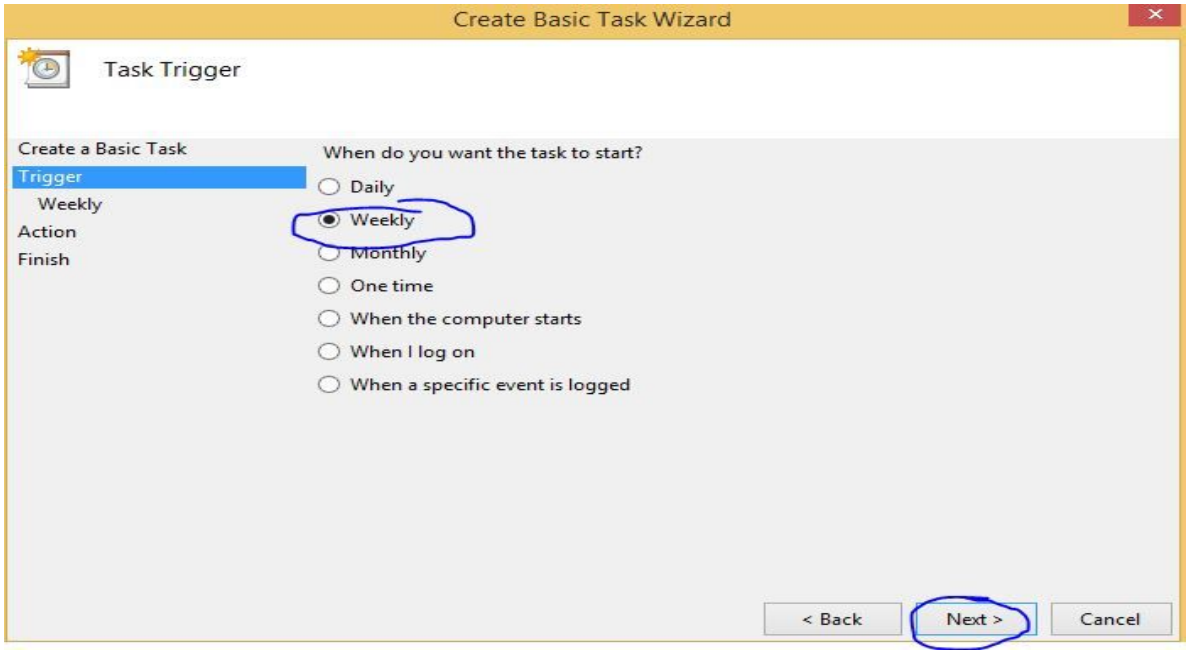
3.3. Click *Create Basic Task*

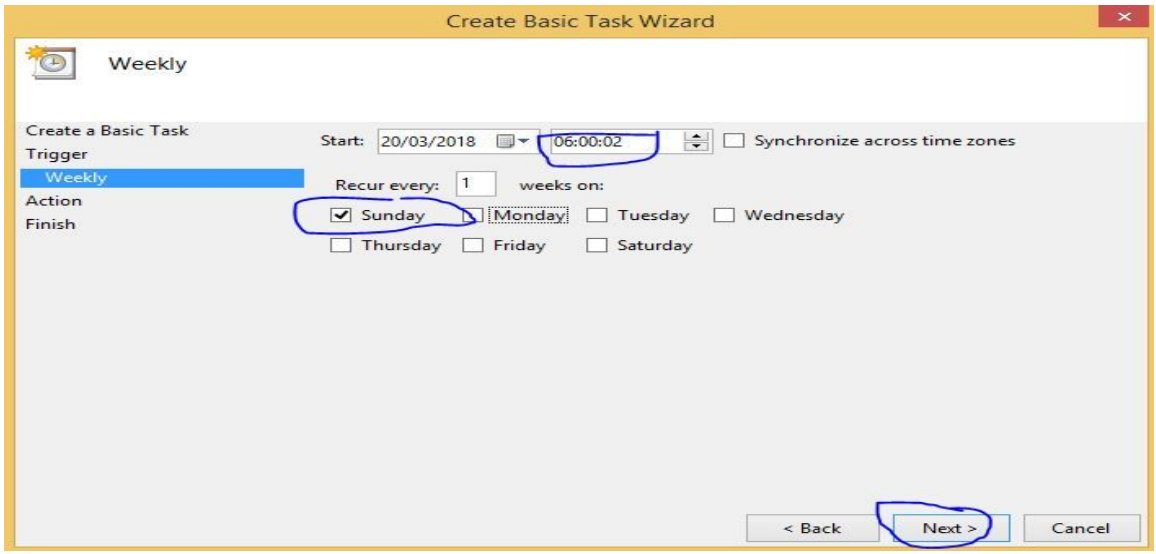


3.4. Give name and description of the task

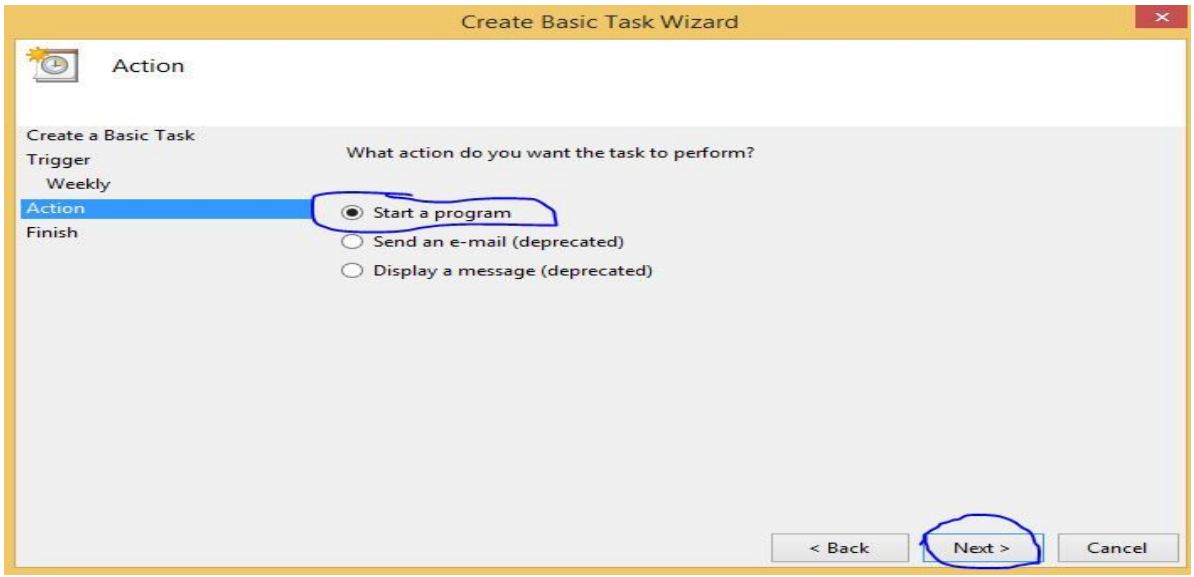


3.5. Define when do you want to start the task





3.6. Select action you want to perform

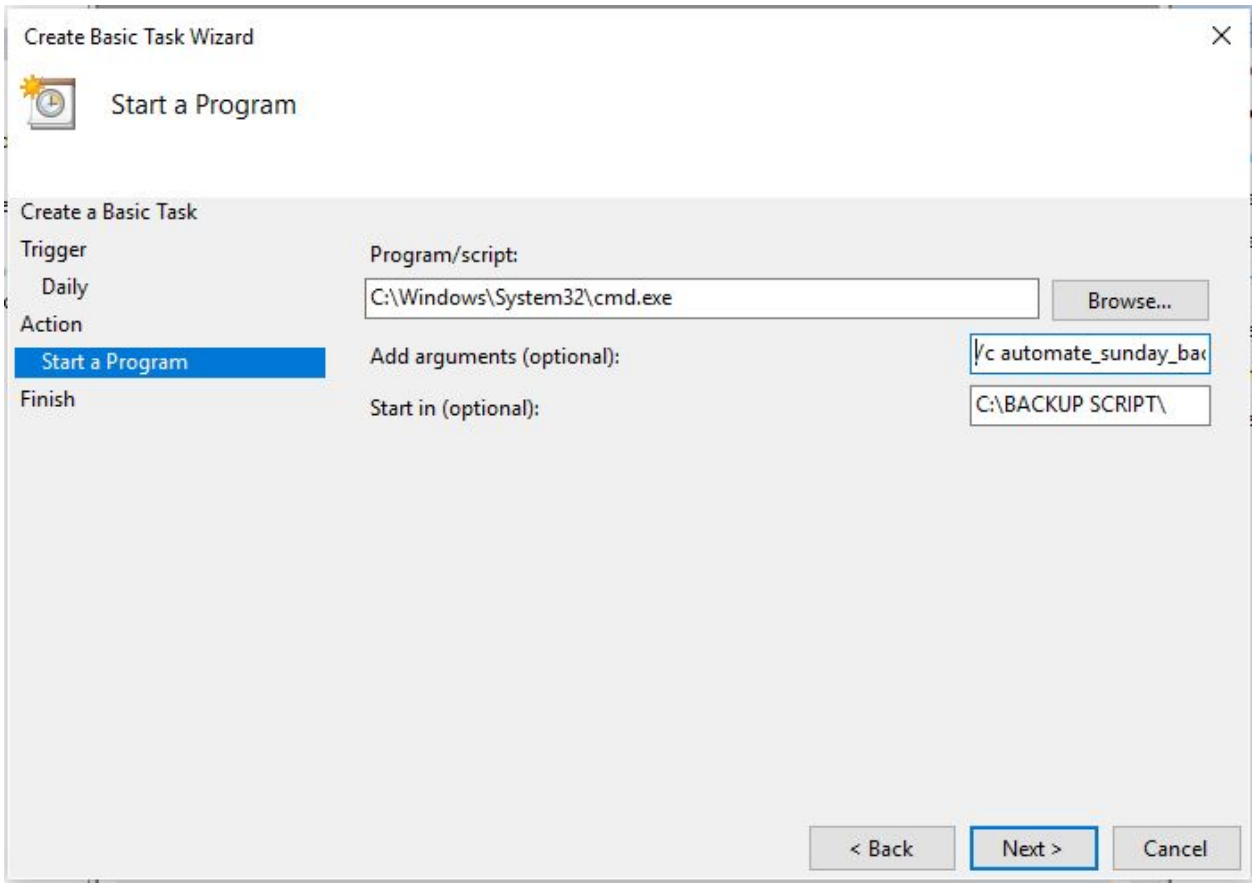


3.7. Select following

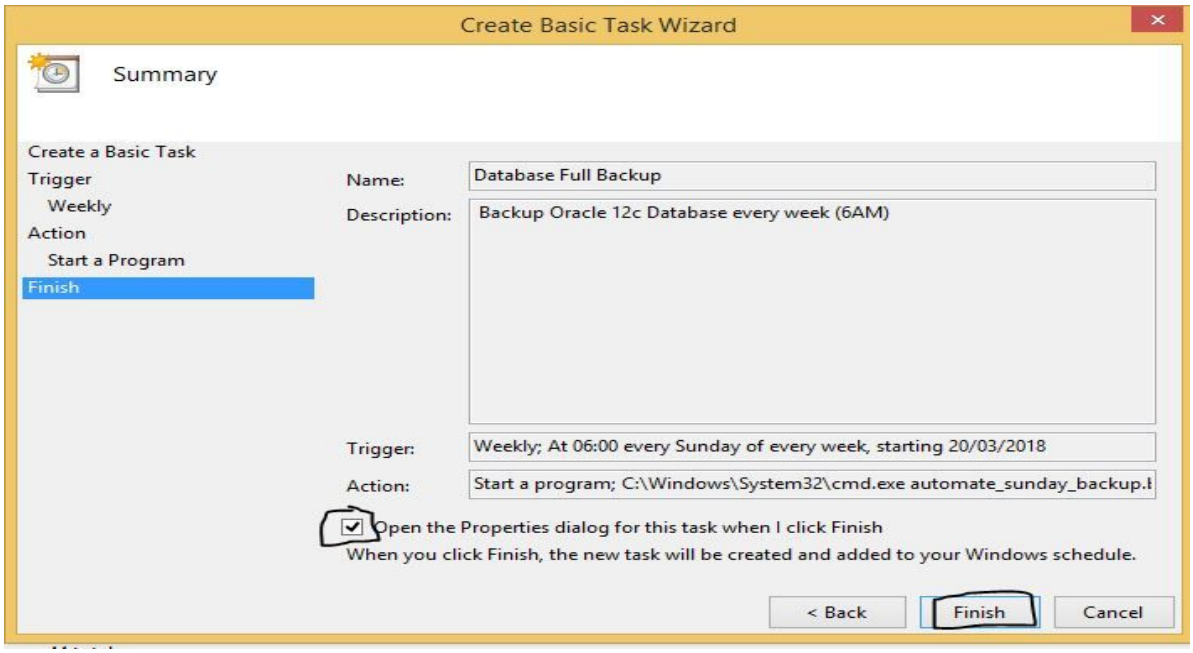
-- In program script select the location of your command prompt(i.e. C:\Windows\system32\cmd.exe)

-- Add .bat file you want to execute as argument(i.e. automate_sunday_backup.bat)

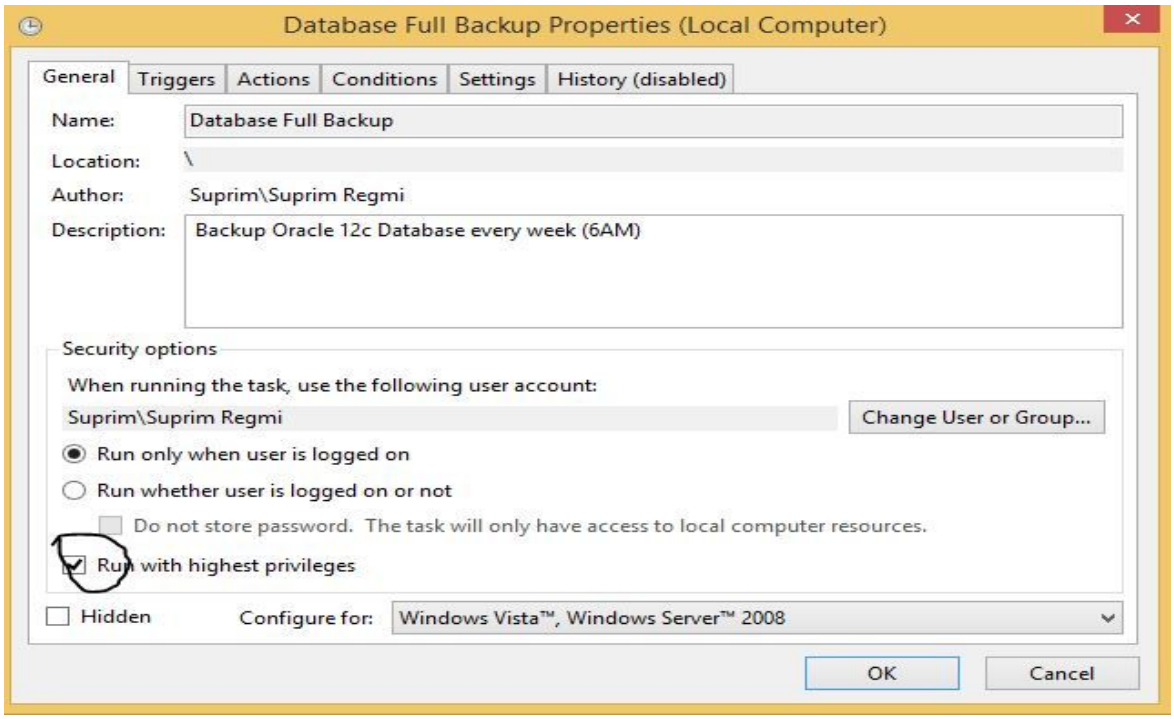
-- Add location of your script In start in field(i.e. C:\BACKUP SCRIPT\)



3.8. Click Finish



3.9. Check the box Run with highest privilege



REFERENCES:

2.4. https://docs.oracle.com/cd/A81042_01/DOC/server.816/a76990/rmansy43.htm

2.5.1 https://docs.oracle.com/cd/B28359_01/backup.111/b28273/rcmsynta043.htm#RCMRF153

2.5.2. http://neeraj-dba.blogspot.com/2011/06/handling-corrupt-datafile-blocks-in.html

2.6. https://docs.oracle.com/cloud/latest/db112/BRADV/rcmbckad.htm#BRADV89569

2.7. http://www.dba-oracle.com/t_alter_system_switch_logfile_vs_alter_system_archivelog_current.htm