

# Shell Scripting: Create Report / Log File Names With Date in Filenames

S ometime you need to create a shell scripts with output filenames with date in it. For example instead of log file named “secure.log”, you can create a filename called “secure-jan-02-06.log” or “secure-DATE-FORMAT.log”. The DATE-FORMAT can be set as per requirements:



=> dd-mm-yyyy  
=> mm-dd-yyyy  
=> yyyy-mm-dd  
=> Month-Day-Year

The date in the filename will make it easy to find out all logs or reports. You can display the current date and time in the given FORMAT using date command. If you just type date command it will display in standard FORMAT:

# UNIX

```
$ date
```

Output:

```
Sun Feb  5 18:23:44 IST 2006
```

## date Command Syntax

The syntax is as follows for GNU/date:

```
date +"FORMAT"
```

To display date in MONTH-DD-YYYY format use the date command as follows:

```
$ date +"%b-%d-%y"
```

Sample outputs:

```
Feb-05-06
```

Where, FORMAT can be any one of the following:

- **%a** : Abbreviated weekday name (Sun..Sat)
- **%b** : Abbreviated month name (Jan..Dec)
- **%B** : Full month name, variable length (January..December)
- **%d** : Day of month (01..31)
- **%e** : Day of month, blank padded ( 1..31)
- **%m** : Month (01..12)
- **%Y** : Year
- **%d** : Day of month (e.g, 01)
- **%H** : 24 hour format (00..23)
- **%I** : 12 hour format (01..12)
- **%j** : day of year (001..366)
- **%D** : date; same as %m/%d/%y
- **%F** : full date; same as %Y-%m-%d (a good format for sorting filename)

Get the date in YYYY-mm-dd format

```
NOW=$(date +"%Y-%m-%d")
```

OR

```
NOW=$(date +"%F")
```

To create a file with date in filename, enter:

```
LOGFILE="log-$(date +%F).log"
```

To display filename, enter:

```
echo "$LOGFILE"
```

## Sample Shell Script

```
#!/bin/sh
# Dump MySQL database every hour using cron
# Author: Vivek Gite
# See for more info: http://www.cyberciti.biz/tips/shell-scripting-creating-reportlog-file-na
```

```
# -----  
  
## date format ##  
NOW=$(date +%F)  
NOWT=$(date +%T)  
  
## Backup path ##  
BAK="/nfs/backup/mysql/$NOW"  
  
## Login info ##  
MUSER="madmin"  
MPASS="PASSWORD"  
MHOST="127.0.0.1"  
  
## Binary path ##  
MYSQL="/usr/bin/mysql"  
MYSQLDUMP="/usr/bin/mysqldump"  
GZIP="/bin/gzip"  
  
## Get database list ##  
DBS="$($MYSQL -u $MUSER -h $MHOST -p$MPASS -Bse 'show databases')"  
  
## Use shell loop to backup each db ##  
for db in $DBS  
do  
    FILE="$BAK/mysql-$db-$NOWT.gz"  
    echo "$MYSQLDUMP -u $MUSER -h $MHOST -p$MPASS $db | $GZIP -9 > $FILE"  
done
```