

# Need to do redundant tasks? Let the shell/batch scripts do that for you!

(all the scripts have been tried in windows 7)

## BATCH FILE (BAT)

1. List folder file (only name) into a textfile

```
dir /b wave > input_list0.txt
```

2. Move to a folder from inside a batchscript (.bat)

```
cd /d e:\__March_speechToPARCOR
```

3. Open cmd and move into working folder

```
cmd /k "cd /d e:"
```

## CYGWIN

1. Open Cygwin to a specific folder

- Open setup file for Cygwin, install (or make sure it has already been installed) chere package under the "Shells" category
- Open Cygwin as administrator, type  

```
chere -i
```
- Now you can have "Bash Prompt Here" in the right-click context menu

## SHELL SCRIPT (SH)

1. List folder file (only name) into a textfile

```
ls wave > input_list0.txt
```

2. Move to a folder from inside a batchscript (.bat)

```
cd e:\__March_speechToPARCOR
```

## PERL

1. Rename (or remove) certain word(s)

```
$your_variable =~ s/previous_word/expected_word/;
```

```
#!/usr/bin/perl
# command: perl rename_extension.pl input_list0.txt > input_list.txt
use strict;
open(IN, $ARGV[0]);
my $line;
my @phone;
while($line = <IN>){
    chomp($line);
    $line =~ s/¥.wav//;
    print "$line¥n";
}
close(IN);
```

make sure to have your End of Line (EOL) in Unix mode

2. Open file, take only the right part of the data (e.g., MSE), rewrite into the other file:  
ps:  $\$s+\$$  represents EOL

```
#!/usr/bin/perl

$input_file = "log.txt";
$result_file = "log_right.txt";

open(RESULT_FILE, ">$result_file") || die "Cannot open result file \n";
open(INPUT_FILE, "$input_file") || die "Cannot open input file \n";
# < or nothing: input, > output, >> append

@input_file_all_file = <INPUT_FILE>;
foreach $input_line (@input_file_all_file){
    @input_line_tab = split(/\t/, $input_line);
    @input_line_tab[$#input_line_tab] =~s/\$s+$/;
    print RESULT_FILE @input_line_tab[$#input_line_tab]."\n";
}

close(INPUT_FILE);
close(RESULT_FILE);
```

3. After getting 1 column MSE, looking for the minimum MSE

```
#!/usr/bin/perl

open(RESULT_FILE, "$result_file") || die "Cannot open result file \n";
@result_file_all_file = <RESULT_FILE>;
$x = @result_file_all_file[0]+0; #add zero to change to number

foreach $result_line (@result_file_all_file){
    $y = $result_line+0;
    $min = ($x, $y)[$x > $y];
    $x = $min;
}

print "minimum MSE=$x";
close(RESULT_FILE);
```

## MATLAB

1. Running matlab scripts without opening matlab GUI
  - add your current directory into MATLAB path

```
% addpwd.m
```

```
function addpwd
% addpwd Add the current directory to the path
oldPath = path;
thisDir = pwd;
path(oldPath, thisDir);
fprintf('The following directory has been added to the path %s\n', thisDir);
```

- **make sure your matlab script run perfectly**

```
% you might wanna put ';' (semicolon) after all the commands
% you might wanna do this: currentDir = pwd;
% and refer all the connected files to currentDir
```

- **Run this command from command prompt**

```
% matlab -nodisplay -nojvm -r yourScript > yourScript.out
matlab -nodisplay -nojvm -r yourScript
```