

## Example Command File

When users are trying to design instruments or create music files, they may want to record the commands necessary to build their creations. This way they can reproduce the work, or edit the commands in order to generate a different version. Command, or script, files are a common way to produce a repeatable sequence of commands.

Script files can be used as input for a program by telling the shell program to use the file contents in place of the user keyboard input. This is called "redirecting input".

For example, if we have commands in a file called `example_commands.txt`, and we want to use the commands in that file as input to the program `program-menu-test/menu_test`, we can do this in the shell like this:

```
$ ./program-menu-test/menu_test < example_commands.txt
```

This would require that the shell is in the top level directory of the code repository, the program had been built correctly, and the `example_commands.txt` file exists in the top level directory. If the file is in a different directory, just provide the correct location. The `<` character indicates that the input should be read from the file.

Here is an example script file: [example\\_commands.txt](#)

```
# This is a sample input file.
# If you use it as input to the menu_test program,
# you'll see a bunch of "Choice? " prompts,
# followed by the output of a few echo
# and help commands.
echo -- Menu test --
echo
echo Hello world!
echo Comments can be generated with # or comment commands
echo
# This a comment.
comment This is also a comment.
echo
echo This is the output of the help command.
help
echo
echo This is the output of the menu command.
menu
echo
echo They look the same, eh?
echo
echo Goodbye world.
quit
echo This command should not run, because we already quit.
```

Download it and use it to see if you get the same output as below:

```
$ program-menu-test/menu_test < example_commands.txt
Choice? Choice? Choice? Choice? Choice? Choice? -- Menu test --
Choice?
Choice? Hello world!
Choice? Comments can be generated with # or comment commands
Choice?
Choice? Choice? Choice?
Choice? This is the output of the help command.
Choice? Options are:
  # - Skip to end of line (comment).
  comment - Skip to end of line (comment).
  echo - Echo back the arguments given.
  help - Display help message.
  menu - Display help message.
  quit - Terminate the program.

Choice?
Choice? This is the output of the menu command.
Choice? Options are:
  # - Skip to end of line (comment).
  comment - Skip to end of line (comment).
```

```
echo - Echo back the arguments given.  
help - Display help message.  
menu - Display help message.  
quit - Terminate the program.
```

```
Choice?
```

```
Choice? They look the same, eh?
```

```
Choice?
```

```
Choice? Goodbye world.
```

```
Choice?
```

```
$
```

You can also make your own script files.