Conditional Commands

When an internal command or external program finished, it returns a result called the exit code. Conditional commands allow you to perform tasks based upon the previous command's exit code. Many programs return a 0 if they are successful and a non-zero value if they encounter an error.

If you separate two commands by **&&** (AND), the second command will be executed only if the first returns an exit code of 0. For example, the following command will only erase files if the BACKUP operation succeeds:

```
[c:\] backup c:\ a: && del c:\*.bak;*.lst
```

If you separate two commands by || (OR), the second command will be executed only if the first returns a non-zero exit code. For example, if the following BACKUP operation fails, then ECHO will display a message:

```
[c:\] backup c:\ a: || echo Error in the backup!
```

All internal commands return an exit code, but not all external programs do. Conditional commands will behave unpredictably if you use them with external programs which do not return an explicit exit code. To determine whether a particular external program returns a meaningful exit code use an **ECHO %?** command immediately after the program is finished. If the program's documentation does not discuss exit codes you may need to experiment with a variety of conditions to see how the exit code changes.

From:
http://185.82.219.184/doku/ - **osFree wiki**

Permanent link:
http://185.82.219.184/doku/doku.php?id=en:docs:cmd:other:cond&rev=1400914486

Last update: **2014/05/24 06:54**