

# ToolMan's Approach to Documenting UNIX Directories

by Daniel E. Singer  
<des@cs.duke.edu>

I think documentation of computer systems is very important, including documentation of systems, methods, applications, and programs, but also of directories and their contents, which is often overlooked.

One problem I run into with UNIX (or any OS, for that matter) is that I end up accumulating hordes of directories full of files that are hard to keep track of, i.e., to know what all the scripts and data files are, or even if they *are* scripts or data files. README files – the usual approach to this problem – are OK (and better than nothing), but they soon get out of date: new files show up (mysteriously!) in the directory, some go away or change their names, and the README file is difficult to maintain to accurately reflect these changes, especially when there are dozens of entries in the directory.

A similar problem occurs, when I need to look for something in one of somebody else's directories, whether it's because they're on the phone, out for the day, have a new job, or got hit by a bus. Usually the only way to tell what things are is to open one file at a time and start reading. You know how much fun that is. Wouldn't it be nice if in some of these directories there were an up-to-date listing that included a short description of what each item in the directory was and perhaps some comments on why the directory is there? Can you say "pipe dream"?

So let's see, documentation is good, directories are an incredible pain to document . . . this calls for a tool! Well, during some of those spare hours at work, and at home nights and weekends, I wrote a tool – a program in Bourne shell – called *check*. The reason it's called *check* is that it checks for consistency between a directory and an INDEX file (that's the default name) in the directory and tells you what the inconsistencies are. It can even create and update the INDEX file, though the process does eventually involve some manual editing.

Time for an example. Let's say there's a directory like the one shown in Listing 1. Files and such have accumulated over several months. We `cd` to the directory and type `check -um` (the "u" tells *check* to update the INDEX file, and the "m" tells it to invoke an editor afterward). The *check* script will give a warning that there's no INDEX file, ask if it should create one, and create a bare bones file with a comment header and directory entries listed, as in Listing 2. It will then invoke our favorite editor on the file for us to add a

description to each file entry, add any other comments, and rearrange it all to our liking, perhaps as in Listing 3.

**Listing 1: Directory Contents Before check Has Been Used**

```
% pwd
/home/des/backups
% ls -A
backup_misc      host_dul.sh
backup_misc1     hosts
backup_storm     hosts.9502
blist            hosts.9503
blist.9502       out1
blist.9503       out2
blist.9504       out5
blist.9504.1     t
du.av.950203     t.sh
du.av.950301     tape-params
du.av.950407     tbk-src
host_du.sh
```

**Listing 2: Initial INDEX File Created by check**

```
#
# @(#) Index to files in /home/des/backups
#
# This file is maintained by the 'check'
# program.
#
INDEX          - Index to files in /home/des/
..             backups (this file)

backup_misc    - ?
backup_misc1   - ?
backup_storm   - ?
blist         - ?
blist.9502    - ?
blist.9503    - ?
blist.9504    - ?
blist.9504.1  - ?
du.av.950203  - ?
du.av.950301  - ?
du.av.950407  - ?
host_du.sh    - ?
host_dul.sh   - ?
hosts        - ?
hosts.9502    - ?
hosts.9503    - ?
out1         - ?
out2         - ?
out5         - ?
t           - ?
t.sh        - ?
tape-params  - ?
tbk-src     - ?
```

**Listing 3: INDEX File After Being Edited**

```
#
# @(#) Index to files in /home/des/backups
#
# This directory contains miscellaneous
# scripts and data relating to system backups.
#
# This file is maintained by the 'check'
# program.
#
INDEX          - Index to files in /home/
..             des/backups (this file)

tbk-src        - dir w/ 'tbackup' and
..             'chkbackup' src

# scripts
backup_misc    - script to do certain
..             specialized backups such as:
..             calendar and crontabs; ltmps;
..             backup log mirroring;
backup_misc1   - old version
backup_storm   - script to backup host storm
..             before was part of reg.
..             backups
host_du.sh     - script to list fs's and blocks
..             of local disks for one or more
..             hosts to assist in backup
..             planning; see opts in source
host_dul.sh    - old version
t.sh          - just testing...

# data
blist         - current backup list of hosts
..             with disk KB summaries
blist.9502    - old blist
blist.9503    - ditto
blist.9504    - ditto
blist.9504.1  - ditto
du.av.950203  - detailed listing of hosts and
..             partition backup info from
..             'hosts_du.sh -a -v -f hosts'
du.av.950301  - ditto
du.av.950407  - ditto
hosts        - list of hosts to backup
hosts.9502    - old list
hosts.9503    - ditto
out1         - output from 'hosts_du.sh -a
..             -s -f hosts'
out2         - ditto
out5         - ditto
t           - output from 't.sh'

# misc
tape-params   - code segments from SunOS 4.1.3
..             'st_conf.c'
```

Now suppose some time has gone by and the directory looks like Listing 4. Again type `check -um`. Assuming that the INDEX file has not been edited in the meantime, the output in Listing 5 will result. The INDEX file at this point looks like Listing 6. Pressing the <Return> key, we enter the editor to add descriptions to the new entries, move things around, and perhaps remove the “deleted” entries (*check* deletes by inserting a comment indicator [there are three different ones] at the beginning of the entry, in this case, a “.”).

#### Listing 4: Revisited Directory

```
% pwd
/home/des/backups
% ls -A
AdvFS          blist.9607    hosts
INDEX          blist.9607s  hosts.9502
SCCS           blist.9607t  hosts.9503
backup_kedem.sh blist.9701.1 hosts.9505
backup_misc    blist.9701t  hosts.9507
backup_storm   du.av.950203 hosts.9601
blist          du.av.950301 hosts.9607
blist+totals.sh du.av.950407 kedem_disk.out
blist.9502     du.av.950706 mondays.sh
blist.9503     du.av.960102 out8
blist.9504     du.av.960627 out9b
blist.9507     du.av.960701 pbk
blist.9507.a   du.av.9701   tape-params
blist.9601     host_du.sh   tbk-src
blist.9601.a   host_dul.sh
```

#### Listing 5: Output From Running `check -um`

```
% check -um

check: running in directory "/home/des/
        backups".

AdvFS
SCCS
backup_kedem.sh
sh blist+totals.sh
blist.9507
blist.9507.a
blist.9601
blist.9601.a
blist.9607
blist.9607s
blist.9607t
blist.9701.1
blist.9701t
du.av.950706
du.av.960102
du.av.960627
du.av.960701
du.av.9701
hosts.9505
hosts.9507
```

```
hosts.9601
hosts.9607
kedem_disk.out
mondays.sh
out8
out9b
pbk
check: 27 omissions found in "INDEX".
backup_misc1
t.sh
blist.9504.1
out1
out2
out5
t
check: 7 extras found in "INDEX".

check: update "INDEX" (y/n)? y
check: "INDEX" updated.

check: edit file "INDEX" (y/n)?
```

#### Listing 6: INDEX File After the Second Run of `check -um`, But Before Editing

```
#
# @(#) Index to files in /home/des/backups
#
# This directory contains miscellaneous
# scripts and data relating to system backups.
#
# This file is maintained by the 'check'
# program.
#

INDEX          - Index to files in /home/
..             des/backups (this file)

tbk-src        - dir w/ 'tbackup' and
..             'chkbackup' src

# scripts

backup_misc    - script to do certain
..             specialized backups such as:
..             calendar and rontabs; ltmps;
..             backup log mirroring;
. backup_misc1- old version
backup_storm   - script to backup host storm
..             before was part of reg.
..             backups
host_du.sh     - script to list fs's and blocks
..             of local disks for one or more
..             hosts to assist in backup
..             planning; see opts in source
host_dul.sh    - old version
. t.sh        - just testing...
```

```

# data
blist          - current backup list of
..             hosts with disk KB
..             summaries
blist.9502     - old blist
blist.9503     - ditto
.blist.9504    - ditto
blist.9504.1   - ditto
du.av.950203   - detailed listing of hosts
..             and partition backup info
..             from 'hosts_du.sh -a -v -f
..             hosts'
du.av.950301   - ditto
du.av.950407   - ditto
hosts          - list of hosts to backup
hosts.9502     - old list
hosts.9503     - ditto
.out1          - output from 'hosts_du.sh
..             -a -s -f hosts'
.out2          - ditto
.out5          - ditto
.t            - output from 't.sh'

# misc

tape-params    - code segments from SunOS
..             4.1.3 'st_conf.c'

AdvFS          - ?
SCCS           - ?
backup_kedem.sh - ?
blist+totals.sh - ?
blist.9507     - ?
blist.9507.a   - ?
blist.9601     - ?
blist.9601.a   - ?
blist.9607     - ?
blist.9607s   - ?
blist.9607t   - ?
blist.9701.1   - ?
blist.9701t   - ?
du.av.950706   - ?
du.av.960102   - ?
du.av.960627   - ?
du.av.960701   - ?
du.av.9701     - ?
hosts.9505     - ?
hosts.9507     - ?
hosts.9601     - ?
hosts.9607     - ?
kedem_disk.out - ?
mondays.sh     - ?
out8           - ?
out9b          - ?
pbk            - ?

```

Voila! Now the INDEX file is completely up to date (i.e., after the manual edit step), and we didn't have to hunt and peck through it or the directory to find what needed to be added or deleted. Normally, in cases of directories with dozens or even hundreds of entries, this would be difficult to impossible.

The *check* script requires a small amount of special formatting of the INDEX file so that it knows which items are entries and which are comments. It even allows for two fairly distinct INDEX file formats to accommodate different tastes or situations (the default format was used in our example). The script has additional features (options) for ignoring certain suffixes (like ".Z"), excluding patterns (e.g., "-x '\* .o'"), interactive editing, and recursion through directories. It also has online help (*check -h*). Alternatively, it can be installed with the name *toc* (Table Of Contents), in which case its behavior becomes somewhat altered (a schizoid script by an O.C. programmer!).

So there you have it, a cool tool to assist in keeping that oft overlooked component of your system – directories – documented. Take it or leave it. If you wish to take it, *check* is available at <ftp://ftp.cs.duke.edu/des/scripts/check> and <http://www.cs.duke.edu/~des/scripts/check>. It's approaching 2,000 lines of excessively commented code, which is why it doesn't appear here as a listing. For discussion or notices of updates, you can subscribe to the *check* list at [majordomo@cs.duke.edu](mailto:majordomo@cs.duke.edu)

