

Systems Administration Toolkit: Log file basics

Skill Level: Intermediate

[Martin Brown \(mc@mcslp.com\)](mailto:mc@mcslp.com)

Professional writer

Freelance

26 Feb 2008

A typical UNIX® or Linux® machine creates many log files during the course of its operation. Some of these contain useful information; others can be used to help you with capacity and resource planning. This article looks at the fundamental information recorded within the different log files, their location, and how that information can be used to your benefit to work out what is going on within your system.

About this series

The typical UNIX administrator has a key range of utilities, tricks, and systems he or she uses regularly to aid in the process of administration. There are key utilities, command line chains, and scripts that are used to simplify different processes. Some of these tools come with the operating system, but a majority of the tricks come through years of experience and a desire to ease the system administrator's life. The focus of this series is on getting the most from the available tools across a range of different UNIX environments, including methods of simplifying administration in a heterogeneous environment.

Log files

All systems generate a varying quantity of log files that track and record different information about your machine. The content and utility of these files varies from system to system, but the core information provided by the files is often consistent.

For example, all UNIX and Linux machines use the syslog, a generic logging system that is used by the operating system and applications and services to log information. The syslog records a whole host of data, including logins, performance

information, and failures reported by different hardware and systems. In addition to the syslog, systems also have a variety of service, environment, and application logs that record information about the machine and its operation.

Although parsing and extracting the content of the log files for information can be time consuming and sometimes complex, the wealth of information in those logs is difficult to ignore. The log file can provide hints on potential problems, faults, security lapses and, if used correctly, can even help provide warnings on load and capacity of your servers.

Log locations

The location of the various log files varies from system to system. On most UNIX and Linux systems the majority of the logs are located in `/var/log`. For example, Listing 1 shows a list of logs located on a Gentoo Linux system.

Listing 1. Linux `/var/log` directory contents

```
$ ll /var/log
total 3312
-rw-r----- 1 root
root      8218
2007-11-03 06:21
dmesg
-rw-rw---- 1
portage portage
650111 2008-02-02
13:01 emerge.log
-rw----- 1 root
root      24024
2007-11-05 07:26
faillog
-rw-r--r-- 1 root
root      386032
2007-09-28 14:39
genkernel.log
drwxr-xr-x 2 root
root      4096
2007-11-03 06:47
iptraf/
-rw-r--r-- 1 root
root      292292
2008-02-03 08:07
lastlog
-rw----- 1 root
root      1346931
2008-02-03 08:50
messages
drwxr-xr-x 2 root
root      4096
2006-08-30 17:04
news/
drwxr-xr-x 3 root
root      4096
2007-09-28 13:22
portage/
drwxrwx--- 2 root
portage   4096
```

```

2007-11-03 06:40
sandbox/
drwxrwx--- 2
snort snort
4096 2007-10-13
11:34 snort/
-rw-rw-r-- 1 root
utmp 496896
2008-02-03 08:07
wtmp
-rw-rw-rw- 1 root
mc 61189
2007-06-10 11:37
Xorg.0.log
-rw-rw-rw- 1 root
root 61189
2007-06-10 10:40
Xorg.0.log.old

```

On Solaris®, IBM® AIX®, and HP-UX®, the main syslog and most of the other logs are written to files within the /var/adm directory (Listing 2).

Listing 2. Traditional UNIX /var/adm contents

```

$ ls -al /var/adm
total 230
drwxrwxr-x 9
root sys
512 Feb 3 15:30
.
drwxr-xr-x 48
root sys
1024 Feb 3 15:32
..
drwxrwxr-x 5
adm adm
512 Feb 2 16:13
acct
-rw----- 1
uucp bin
0 Jan 12 18:49
aculog
drwxr-xr-x 2
adm adm
512 Feb 2 16:03
exacct
-r--r--r-- 1
root root
2856 Feb 3 16:10
lastlog
drwxr-xr-x 2
adm adm
512 Feb 2 16:03
log
-rw-r--r-- 1
root root
69065 Feb 3
16:08 messages
drwxr-xr-x 2
root sys
512 Feb 2 16:09
pool
drwxrwxr-x 2
adm sys

```

```

512 Feb  2 16:13
sa
drwxr-xr-x  2
root      sys
512 Feb  2 17:03
sm.bin
-rw-rw-rw-  1
root      bin
0 Jan 12 18:47
spellhist
drwxr-xr-x  2
root      sys
512 Feb  2 16:03
streams
-rw-----  1
root      root
93 Feb  3 16:08
sulog
-rw-r--r--  1
root      bin
3720 Feb  3 16:14
utmpx
-rw-r--r--  1
adm       adm
29760 Feb  3
16:10 wtmpx

```

In addition, some non-system-level messages and information are written into logs located within `/var/log` (Listing 3). For example, on Solaris, by default, mail debug entries are written into `/var/log/syslog`.

Listing 3. Additional logs in `/var/log` on Solaris

```

$ ls -al
/var/log/
total 48158
drwxr-xr-x  7
root      sys
512 Feb  3 16:07
.
drwxr-xr-x 48
root      sys
1024 Feb  3 15:32
..
-rw-----  1
root      sys
0 Jan 12 18:48
authlog
-rw-r--r--  1
root      other
27 Feb  2 16:17
brlog
drwxr-xr-x  2
root      root
512 Feb  2 16:39
gdm
drwxr-xr-x  2
root      sys
512 Feb  2 16:09
pool
-rw-r--r--  1
root      sys
24480410 Feb  3
12:51 postrun.log

```

```
drwxr-xr-x  2
root      sys
512 Feb  2 16:41
swupas
-rw-r--r--  1
root      other
635 Feb  2 17:25
sysidconfig.log
-rw-r--r--  1
root      sys
3967 Feb  3 16:08
syslog
drwxr-xr-x  3
root      sys
512 Feb  2 17:25
webconsole
drwxr-xr-x  2
root      sys
512 Feb  2 16:37
xen
-rw-r--r--  1
root      root
66171 Feb  3
16:07 Xorg.0.log
-rw-r--r--  1
root      root
66256 Feb  3
16:06
Xorg.0.log.old
```

Of course finding the files is the least of the issues. You need to know what the files contain for the information to be of any use.

Depending on the UNIX variants, some logs may be littered about in other places, but there has been a significant attempt to standardize on log file locations to one of the directories already mentioned.

Log types and data

Log types fall into two categories, text log files that contain messages and information in a simple text format, and files that are encoded in a binary format. The former is used for most of the logs in your typical system as they are easy to write and, perhaps more importantly, easy to read. The issue with text files is that they can sometimes be difficult to extract information from in a structured way, because the text format of the files allows the information to be written in any way or structure.

The latter format is more practical for very structured information, or for information that needs to be written in a particular way or format. For the example, the utmp and wtmp data is written to a file in fixed blocks of binary data so that the information can read and be written in a quick and efficient format. Unfortunately, this means that the information is difficult to read without using special tools.

System logs (syslog)

The syslog service is a daemon that runs the background and accepts log entries and writes them to one or more individual files. All messages reported to syslog are tagged with the date, time, and hostname, and it's possible to have a single host that accepts all of the log messages from a number of hosts, writing out the information to a single file.

Messages are also identified by the service that raise the issue (for example, mail, dhcp, kernel), and a class indicating the severity of the message. The severity can be marked as *info* (purely for information), *warning*, *error*, *critical* (a serious problem that needs addressing), and even *emergency* (the system needs urgent help).

The service is highly configurable (generally through `/etc/syslog.conf`, or the equivalent), and allows you to select what classes of information to log, and where to log the information. For example, you can write all the standard information out to a file. But for critical messages, where administrators need the information right away, these messages can be sent immediately to the console. Listing 4 shows the main configuration content of the default `syslog.conf` file from a Solaris 10 installation.

Listing 4. Sample `syslog.conf` file

```
*.err;kern.notice;auth.notice
/dev/sysmsg
*.err;kern.debug;daemon.notice;mail.crit
/var/adm/messages

*.alert;kern.err;daemon.err
operator
*.alert
root

*.emerg
*

...

mail.debug
ifdef('LOGHOST',
/var/log/syslog,
@loghost)

...

ifdef('LOGHOST',
,
user.err
/dev/sysmsg
user.err
/var/adm/messages
user.alert
'root, operator'
user.emerg
*
)
```

Because syslog is a standard logging mechanism within UNIX/Linux it is used to record a massive array of different information. This includes boot messages, login

and authorization information, and service startup/shutdown. In addition, syslog is often used to record e-mail messages delivery, filesystem issues, and even DHCP leases, DNS issues, and NFS problems. Because syslog can write the data to different areas, it's not always obvious that syslog is writing the information.

The main destination for the on-disk copy of the syslog differs between UNIX variants. Many Linux solutions write the information to `/var/log/messages`. On AIX, Solaris, and HP-UX, the syslog is written to `/var/adm/messages`.

You can see a sample of `/var/adm/messages` from a Solaris machine in Listing 5.

Listing 5. Sample system log output

```
Feb  3 16:06:58
solaris2 ata: [ID
496167 kern.info]
cmdk2 at atal
target 0 lun 0
Feb  3 16:06:58
solaris2 genunix:
[ID 936769
kern.info] cmdk2
is
/pci@0,0/pci-ide@1f,1/ide@1/cmdk@0,0
Feb  3 16:06:59
solaris2 asy: [ID
267298
kern.notice]
asy0: UART @
3f8 scratch
register:
expected 0x5a, g
ot 0xff
Feb  3 16:06:59
solaris2 asy: [ID
702181
kern.notice]
Cannot identify
UART chip at 3f8
Feb  3 16:06:59
solaris2 asy: [ID
267298
kern.notice]
asy1: UART @ 2f8
scratch register:
expected 0x5a,
got 0xff
Feb  3 16:06:59
solaris2 asy: [ID
702181
kern.notice]
Cannot identify
UART chip at 2f8
Feb  3 16:07:01
solaris2 genunix:
[ID 314293
kern.info] device
pciclass,030000@2(display#0)
keeps up device
sd@1,0(sd#1), but
the latter is
not
```

```
power managed
Feb  3 16:07:01
solaris2
/usr/lib/power/powerd:
[ID 387247
daemon.error]
Able to open
/dev/srn
Feb  3 16:07:08
solaris2
/sbin/dhcpagent[164]:
[ID 778557
daemon.warning]
configure_v4_lease:
no IP broadcast
specified for
ni0, making best
guess
Feb  3 16:07:31
solaris2
sendmail[503]:
[ID 702911
mail.crit] My
unqualified host
name
(solaris2)
unknown; sleeping
for retry
Feb  3 16:07:32
solaris2
sendmail[507]:
[ID 702911
mail.crit] My
unqualified host
name
(solaris2)
unknown; sleeping
for retry
Feb  3 16:07:48
solaris2
svc.startd[7]:
[ID 652011
daemon.warning]
svc:/system/webconsole:console:
Method
"/lib/svc/method/svc-webconsole
start"
failed with exit
status 95.
Feb  3 16:07:48
solaris2
svc.startd[7]:
[ID 748625
daemon.error]
system/webconsole:console
failed fatally:
transitioned to
maintenance
(see 'svcs -xv'
for details)
Feb  3 16:07:55
solaris2 pseudo:
[ID 129642
kern.info]
pseudo-device:
devinfo0
Feb  3 16:07:55
solaris2 genunix:
[ID 936769
```

```
kern.info]
devinfo0 is
/pseudo/devinfo@0
Feb 3 16:08:31
solaris2
sendmail[503]:
[ID 702911
mail.alert]
unable to qualify
my own domain
name (solaris2)
-- using short
name
Feb 3 16:08:32
solaris2
sendmail[507]:
[ID 702911
mail.alert]
unable to qualify
my
own domain name
(solaris2) --
using short name
```

You can see in the sample output that there is a wide range of information here, from problems and issues with hardware devices through to issues with the current configuration of the mail service.

The format of the file is quite straightforward: it contains the date, hostname, service name, a unique ID (to enable the system to log multi-line messages and have them identified), and the identifier and class of the entry. The remaining text on each line is just free-form text from the system logging the error message.

The format of the file makes it easy to pull out the information you want. All the lines in the file are tagged with a unique ID and all lines are tagged with the identifier and class of the error message.

For example, you can pull out information on critical issues with the mail system by using `grep` to pick out the entries tagged with `mail.crit`: `$ grep mail.crit /var/adm/messages`.

To process the detail of the individual lines within the log is more complex. Although the first few columns within the file are standardized (they are written by the `syslog` daemon), the format of the remainder of the line is entirely dependent on the component reporting the error message.

This can make it complex to read and parse the contents of the file, as you will need to treat each line according to the identifier and reporter. Even then, some lines will not follow a format.

Kernel log (dmesg and alog)

All UNIX and Linux systems have a log that is actually part of the kernel. In practice the log is actually a section of memory in the kernel used to record information about

the kernel that may be impossible to write to disk because the information is generated before the filesystems are loaded.

For example, during the boot process, the filesystems are not accessible for writing (most kernels boot with the filesystem in read mode until the system is considered safe enough to switch to read/write mode). The data in this log contains information about the devices connected to the system and any faults and problems recorded by the system during the boot and operational process.

On some systems the information is periodically dumped into a file (`/var/log/dmesg`); on others it is only available by using the `alog` command (AIX) or `dmesg` (all other UNIX/Linux variants).

The information generated by the kernel is not always written out to another file, such as `syslog`. This can mean that certain pieces of information, such as internal data on devices and hardware, is only available through the `dmesg` log.

For example, Listing 6 shows some sample output from `dmesg` on a Gentoo Linux system. Here it is showing the main boot information, trimmed for brevity.

Listing 6. The `dmesg` log contents

```
$ dmesg
Linux version
2.6.22-gentoo-r8
(root@gentoo2.vm)
(gcc version
4.1.2
(Gentoo 4.1.2
p1.0.1)) #1 SMP
Fri Sep 28
14:22:07 GMT 2007
BIOS-provided
physical RAM map:
  BIOS-e820:
0000000000000000
-
0000000000009fc00
(usable)
  BIOS-e820:
0000000000100000
-
0000000020000000
(usable)
0MB HIGHMEM
available.
512MB LOWMEM
available.
Entering
add_active_range(0,
0, 131072) 0
entries of 256
used
Zone PFN ranges:
  DMA
0 -> 4096
  Normal
4096 -> 131072
```

```
HighMem
131072 ->
131072
early_node_map[1]
active PFN ranges
  0:      0
-> 131072
On node 0
totalpages:
131072
  DMA zone: 32
pages used for
memmap
  DMA zone: 0
pages reserved
  DMA zone: 4064
pages, LIFO
batch:0
  Normal zone:
992 pages used
for memmap
  Normal zone:
125984 pages,
LIFO batch:31
  HighMem zone: 0
pages used for
memmap
DMI not present
or invalid.
Allocating PCI
resources
starting at
30000000 (gap:
20000000:e0000000)
Built 1
zonelists. Total
pages: 130048
Kernel command
line:
root=/dev/ram0
init=/linuxrc
ramdisk=8192
real_root=/dev/hda3
udev
Local APIC
disabled by BIOS
-- you can enable
it with "lapic"
mapped APIC to
ffffd000
(0140c000)
Enabling fast FPU
save and
restore... done.
Enabling unmasked
SIMD FPU
exception
support... done.
Initializing
CPU#0
CPU 0 irqstacks,
hard=c054e000
soft=c052e000
PID hash table
entries: 2048
(order: 11, 8192
bytes)
Detected 2295.874
MHz processor.
```

```
Console: colour
VGA+ 80x25
Dentry cache hash
table entries:
65536 (order: 6,
262144 bytes)
Inode-cache hash
table entries:
32768 (order: 5,
131072 bytes)
Memory:
511616k/524288k
available (3150k
kernel code,
12100k reserved,
818k data, 264k
init, 0k highmem)
virtual kernel
memory layout:
  fixmap :
0xffe17000 -
0xffff0000
(1952 kB)
  pkmap :
0xff800000 -
0xffc00000
(4096 kB)
  vmalloc :
0xe0800000 -
0xff7fe000 (
495 MB)
  lowmem :
0xc0000000 -
0xe0000000 (
512 MB)
  .init :
0xc04e7000 -
0xc0529000 (
264 kB)
  .data :
0xc0413884 -
0xc04e0364 (
818 kB)
  .text :
0xc0100000 -
0xc0413884
(3150 kB)
Checking if this
processor honours
the WP bit even
in supervisor
mode... Ok.
Calibrating delay
using timer
specific
routine.. 4674.89
BogoMIPS
(lpj=23374475)
Mount-cache hash
table entries:
512
CPU: After
generic identify,
caps: 0f80b9b9
00000000 00000000
00000000 00000001
00000000 00000000
CPU: L1 I cache:
32K, L1 D cache:
```

```
32K
CPU: L3 cache:
4096K
CPU: After all
inits, caps:
0f80b9b9 00000000
00000000 00000140
00000001
00000000 00000000
...
```

Listing 7 shows the output from another machine running Gentoo Linux, and in this example you can see some faults being reported by a running filesystem.

Listing 7. Disk error from dmesg

```
EXT3-fs: mounted
filesystem with
ordered data
mode.
sd 7:0:1:0: [sdf]
Result:
hostbyte=0x00
driverbyte=0x08
sd 7:0:1:0: [sdf]
Sense Key : 0x3
[current]
sd 7:0:1:0: [sdf]
ASC=0x4b ASCQ=0x0
end_request: I/O
error, dev sdf,
sector 894959703
EXT3-fs error
(device sdf1):
ext3_get_inode_loc:
unable to read
inode block -
inode=55935010,
block=111869955
sd 7:0:1:0: [sdf]
Result:
hostbyte=0x00
driverbyte=0x08
sd 7:0:1:0: [sdf]
Sense Key : 0x3
[current]
sd 7:0:1:0: [sdf]
ASC=0x4b ASCQ=0x0
end_request: I/O
error, dev sdf,
sector 894959703
```

From [Listing 7](#), you can see that you probably need to check the filesystem, as there appears to be a fault on the filesystem or disk.

In this instance, the information was also reported in the syslog ([Listing 8](#)).

Listing 8. Disk error in syslog

```
messages:Feb  3
12:17:53 bear sd
7:0:1:0: [sdf]
Result:
hostbyte=0x00
driverbyte=0x08
messages:Feb  3
12:17:53 bear sd
7:0:1:0: [sdf]
Sense Key : 0x3
[current]
messages:Feb  3
12:17:53 bear sd
7:0:1:0: [sdf]
ASC=0x4b ASCQ=0x0
messages:Feb  3
12:17:53 bear
end_request: I/O
error, dev sdf,
sector 894959703
messages:Feb  3
12:17:53 bear
EXT3-fs error
(device sdf1):
ext3_get_inode_loc:
unable
to read inode
block -
inode=55935014,
block=111869955
```

But in the case of a serious fault or failure, dmesg can sometimes be your only good source of information on what is happening on your system.

User records (utmp/x, wtmp/x, lastlog)

These files contain the user login and system data logs. Information in these files is written in the special utmp format and so you will need special tools to extract the information.

The data held within these logs records login times and system startup/shutdown times, both for a historical record of the logins and for quick access to the last boot or login time used during login.

See [Resources](#) for another article within the System Administration Toolkit that contains information on how to parse these files.

The cron logs

The cron time daemon, which is responsible for running many services in the background at periodic intervals, generates its own logs of information.

On some systems, the cron log is recorded using syslog, but on Solaris and some traditional UNIX variants, the information is written to the file `/var/cron/log`. The

information contained in the log includes the details of the command executed and when the job started and stopped.

For an example of the log contents, see Listing 9.

Listing 9. The log of cron activity

```
! *** cron
started *** pid
= 283 Sun Feb 3
16:07:10 2008
> CMD:
/usr/local/bin/logmanage
>/dev/null 2>&1
> root 946 c Sun
Feb 3 17:10:00
2008
< root 946 c Sun
Feb 3 17:10:00
2008
> CMD:
/usr/local/bin/backup
>/dev/null 2>&1
> root 949 c Sun
Feb 3 17:11:00
2008
< root 949 c Sun
Feb 3 17:11:01
2008
```

Parsing the contents of the log can be an effective way to determine any problems with jobs that don't seem to execute properly. It can also be a good way to check the execution time of a job. Long-running jobs, or jobs that never seem to have finished, probably indicate a problem that should be investigated.

Log file management

You should make sure that you manage the logs on your systems. Log files can grow very large and in many cases you will want to keep an historical record of events on your machine for problems.

For example, a phantom reboot or shutdown of a system should be investigated, and often the system logs are the only source of information. Although it cannot tell you everything that was taking place at the time the failure occurred, you may get information that helps, such as the precise time of the failure, or information about events that led up to the problem. Potential security problems and login attempts may indicate that your machine was being hacked and that may have led to or even been the cause of the problem.

Keeping months and months of logs is probably not necessary (although it may under some circumstances be a legal requirement). On a busy system you can

easily record 25MB or more information each day to the system logs, and logs are frequently the cause of insufficient disk space errors.

Some UNIX/Linux variants include an automatic log management process (Solaris includes the `/usr/sbin/logadm` command), but it is not that difficult to create your own. A typical arrangement is to keep individual logs for a short period of time (for example, four weeks) and number them sequentially. For example, if you have the file `messages`, last week's file is in `messages.1`, the two-week-old file is in `messages.2`, and so on. This makes migration of the files very easy.

You must, however, be careful that you can successfully copy and recreate the file so that you do not lose any significant amount of information in the migration and archiving process. For the old files, to save space, you can also archive the content. Listing 10 shows a simple script that will copy and archive individual files into a suitably named directory within the original location.

Listing 10. Simple log-archiving facility

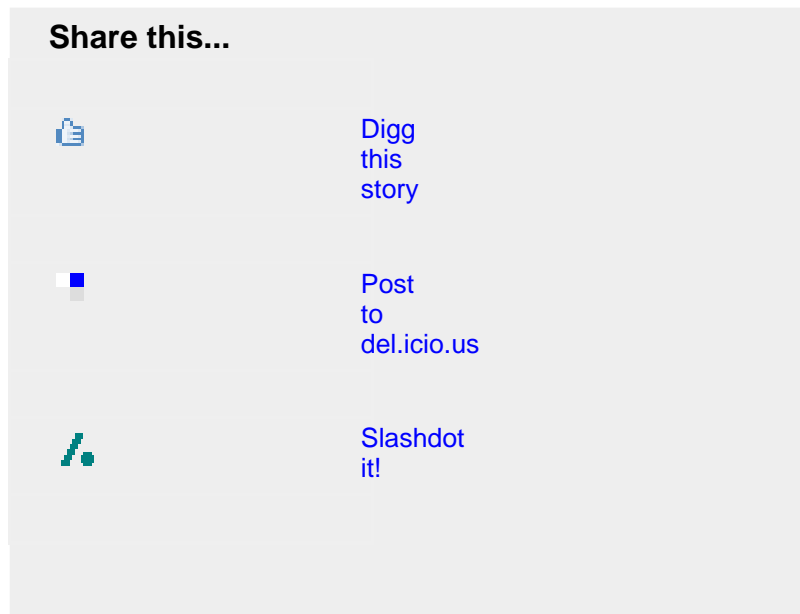
```
#!/bin/bash

# Manage logs and
# archive them if
# necessary
# Keeps 4 copies
# of logs

cd /var/log
for type in cyrus
dmesg emerge.log
faillog
genkernel.log
messages
do
    mkdir -p
    $type.d
    cp
    $type.d/$type.3.bz2
    $type.d/$type.4.bz2
    cp
    $type.d/$type.2.bz2
    $type.d/$type.3.bz2
    cp
    $type.d/$type.1.bz2
    $type.d/$type.2.bz2
    cp $type
    $type.d/$type.1
    && cat </dev/null
    >$type
    bzip2
    -vf9
    $type.d/$type.1
done
```

Running the script copies, recreates, and archives the log files. Note how the files are migrated; we just move the current to the week older in each case. Then finally we archive and recreate the original file.

Summary



Creating new AD users and groups from AIX

Log files can contain a whole wealth of information, but understanding the depth of the information and the format of the files helps immensely when you are trying to diagnose and resolve problems. This article has looked at the basics of log files, their location, and the details of the contents of those files and how they can help you diagnose problems and identify issues before they become problems. The article also examined the format of the different files and the relationships between different files and their contents.

Resources

Learn

- [System Administration Toolkit: Monitoring User Usage](#) (Martin Brown, developerWorks, October 2007) looks at ways of examining the utmp and wtmp files for information about users and their activities.
- [System Administration Toolkit: Monitoring Mail Usage](#) (Martin Brown, developerWorks, December 2007) gives examples on monitoring the system log for mail deliveries and information.
- [System Administration Toolkit: Monitoring Disk Usage](#) (Martin Brown, developerWorks, June 2006) covers many techniques for investigating the disk space used on your machine.
- Read [System Administration Toolkit: Standardizing your UNIX command-line tools](#) (Martin Brown, developerWorks, May 2006) to learn how to use the same command across multiple machines.
- [System Administration Toolkit: Time and event management](#) (Martin Brown, developerWorks, May 2006) covers the creation and organization of time scripts using cron and at.
- For an article series that will teach you how to program in bash, see [Bash by example, Part 1: Fundamental programming in the Bourne again shell \(bash\)](#) (Daniel Robbins, developerWorks, March 2000), [Bash by example, Part 2: More bash programming fundamentals](#) (Daniel Robbins, developerWorks, April 2000), and [Bash by example, Part 3: Exploring the ebuild system](#) (Daniel Robbins, developerWorks, May 2000).
- [System Administration Toolkit](#): Check out other parts in this series.
- [Making UNIX and Linux work together](#) (Martin Brown, developerWorks, April 2006) is a guide to getting traditional UNIX distributions and Linux working together.
- Different systems use different tools, and the IBM Redbook [Solaris to Linux Migration: A Guide for System Administrators](#) will help you identify some key tools.
- [Exploring the Linux memory model](#) (Vikram Shukla, developerWorks, January 2006) helps you understand how Linux uses memory, swap space and exchanges pages and processes between the two.
- [New to AIX and UNIX](#): Visit the New to AIX and UNIX page to learn more about AIX and UNIX.
- The [developerWorks AIX and UNIX zone](#) hosts hundreds of informative articles and introductory, intermediate, and advanced tutorials.

- [AIX Wiki](#): A collaborative environment for technical information related to AIX.
- Stay current with [developerWorks technical events and webcasts](#).
- [Technology bookstore](#) Browse this site for books and other technical topics.

Get products and technologies

- [syslog-ng](#) is an open source implementation of the syslog service that includes improvements and enhancements to make it a more general purpose logging mechanism

Discuss

- Participate in the AIX and UNIX forums:
 - [AIX 5L -- technical forum](#)
 - [AIX for Developers Forum](#)
 - [Cluster Systems Management](#)
 - [IBM Support Assistant](#)
 - [Performance Tools -- technical](#)
 - [Virtualization -- technical](#)
 - [More AIX and UNIX forums](#)

About the author

Martin Brown

Martin Brown has been a professional writer for over eight years. He is the author of numerous books and articles across a range of topics. His expertise spans myriad development languages and platforms -- Perl, Python, Java, JavaScript, Basic, Pascal, Modula-2, C, C++, Rebol, Gawk, Shellscript, Windows, Solaris, Linux, BeOS, Mac OS/X and more -- as well as Web programming, systems management and integration. Martin is a regular contributor to ServerWatch.com, LinuxToday.com and IBM developerWorks, and a regular blogger at Computerworld, The Apple Blog and other sites, as well as a Subject Matter Expert (SME) for Microsoft. He can be contacted through his Web site at <http://www.mcslp.com>.

Trademarks

IBM and AIX are registered trademarks of IBM Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.