Sample Events Files

This appendix contains a sample Events testtab file and some examples of using the Events Command Line Interface (EventsCli) to send alarms to PEP, SNMP, etc. You may also want to refer to Appendix F, "Events Commands," for more details on using the syntax for the testtab and EventsCli Events commands.

Sample testtab Configuration File

The following is an example `testtab` file with many of the optional test entries.

```
#
# This is an example testtab file. Not all test may be on all
# systems. Normally this file is automatically created. Changes
# to this file should be made using the GUI or via SNMP (with your
# network management app). However you may also edit this file.
# When editing this file, you may add tests anywhere, there is
# no specific order required. When Events updates this file, it
# WILL reorder the tests and any comments you added will be lost.
#

*********
  cache
**********
ncache | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:\
   :!pep:severe=3:!log:high=50.00:units=percent:delta=5.00:\
   :+rate=10.00:\
   :/* end  ncache */:
*********
  cpu
**********
cpu load | :\
   :on:testfreq=1:alarmfreq=1:mailer=/bin/mail:notify=jerry:pep:\
   :severe=2:!log:high=5.00:units=units:delta=0.00:\
   :/* end  cpu load */:
cpu idle | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=jerry:\
   :!pep:severe=3:log:units=percent:delta=1:\
   :/* end  cpu idle */:
cpu kernel | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=jerry:pep:\
   :severe=3:log:high=70:units=percent:delta=1:\
   :/* end  cpu kernel */:
cpu user | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=jerry:pep:\
   :severe=3:log:high=70:units=percent:delta=1:\
   :/* end  cpu user */:
```

```
cpu wait | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=jerry:pep:\
   :severe=3:log:high=70:units=percent:delta=1:\
   :/* end  cpu wait */:
*********
  proc
**********
proc_slots | :\
   :on:testfreq=1:alarmfreq=60:mailer=/bin/mail:notify=root:pep:\
   :severe=3:!log:low=98:units=slots:delta=15:\
   :/* end  proc_slots */:
# restart a process if it dies
!nfsd instances | :\
   :on:mailer=/bin/mail:notify=root:pep:\
   :severe=3:log:low=3:high=5:command=/scripts/nfsd_restart:
   :/* end !nfsd */
# alarm if a process starts taking up too much memory,
# maybe it has a memory leak.
!myprog size |:\
   :on:high=500:+jump=25:notify=alice:
   /* end !myprog size */
# alarm if a process becomes a cpu hog
!his_prog time |:\
   :on:high=200:log:delta=10:\
   :/* end !his_prog time */:
*********
  fs
**********
/ blocks free | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=ted:\
   :!pep:severe=2:log:low=48079:units=blocks:delta=100:\
   :/* end  / blocks free */:
/ inodes free | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=ted:\
   :!pep:severe=2:log:low=12083:units=inodes:delta=0:\
   :/* end  / inodes free */:
/usr blocks free | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=ted:\
   :!pep:severe=2:log:low=48079:units=blocks:delta=100:\
   :/* end  /usr blocks free */:
/usr inodes free | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=ted:\
   :!pep:severe=2:log:low=12083:units=inodes:delta=0:\
   :/* end  /usr inodes free */:
```

```
/var blocks free | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=ted:\
   :!pep:severe=2:log:low=9595:units=blocks:delta=0:\
   :/* end  /var blocks free */:
/var inodes free | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=ted:\
   :!pep:severe=2:log:low=2643:units=inodes:delta=0:\
   :/* end  /var inodes free */:
/tmp blocks free | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=ted:\
   :!pep:severe=2:log:low=4085:units=blocks:delta=0:\
   :/* end  /tmp blocks free */:
/tmp inodes free | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=ted:\
   :!pep:severe=2:log:low=612:units=inodes:delta=0:\
   :/* end  /tmp inodes free */:
/opt blocks free | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=ted:\
   :!pep:severe=2:log:low=76969:units=blocks:delta=0:\
   :/* end  /opt blocks free */:
/opt inodes free | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=ted:\
   :!pep:severe=2:log:low=19257:units=inodes:delta=0:\
   :/* end  /opt inodes free */:
# The /SWAP is a psuedo name and refers to total system
# swap space regardless of the actual number of swap devices.
/SWAP blocks free | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=ted:pep:\
   :severe=5:log:low=2:units=blocks free:delta=2000:\
   :/* end  /SWAP blocks free */:
*********
  files
**********
#  The size test could be used to automatically archive the many
# system and other log files that can grow without bound.
/var/adm/myapp.log size | :\
   :on:testfreq=60:alarmfreq=60:mailer=/bin/mail:notify=root:pep:\
   :severe=2:log:units=size:delta=0:high=200000:\
   :command=~/policy/archive.sh:
   :/* end  /var/adm/sulog size */:
# Alarm each time a file is accessed.
/secure_data accessed |:\
   :on:testfreq=1:alarmfreq=1:\
   :-jump=1:+jump=1:pep:delta=1:log:
   :/* end /secure_data accessed */
```

```
# Alarm each time a file is modified
/mystuff/setuid_script modified |:\
   :on:testfreq=1:alarmfreq=1:\
   :-jump=1:+jump=1:
   :/* end /mystuff/setuid_script modified */
*********
  QUEUES
*********
# This group is good for detecting when an email, printer, or
# other directory based queue is getting backed up.

#This test alarms when there are two many files in the directory.
# It might be used to alart you when it's time to start shifting
# print jobs to another printer.
/usr/spool/lp/laser24 |:\
   :on:testfreq=5:alarmfreq=30:\
   :log:delta=10:\
   :high=100:\
   :/* end /usr/spool/lp/laser24 */:
#This test is usefull when care about how many 'old' files are
# stuck in the queue. It is like the above test, but contains a
# value for 'age' (minutes).
# Specifically, this test will alarm when the mail queue has more
# than 20 files that are more than 30 minutes old.
/usr/spool/mqueue |:\
   :on:testfreq=5:alarmfreq=30:command=/usr/gordon/vickers:\
   :age=30:high=20:\
   :/* end /usr/spool/mqueue */:
*********
  printers
**********
printers | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:\
   :!pep:severe=3:!log:units=....:delta=0:\
   :/* end  printers */:
*********
  rpc
**********
rpcc_calls | :\
   :off:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:\
   :!pep:severe=3:!log:units=calls:delta=100:\
   :/* end  rpcc_calls */:
```

```
rpcc_badcalls | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:\
   :!pep:severe=3:!log:units=bad calls:delta=30:+jump=10:\
   :/* end  rpcc_badcalls */:
rpcc_retrans | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:\
   :!pep:severe=3:log:units=retransmissions:delta=30:+jump=10:\
   :/* end  rpcc_retrans */:
rpcc_badxid | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:\
   :!pep:severe=3:log:units=units:delta=0:+jump=10:\
   :/* end  rpcc_badxid */:
rpcc_timeout | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:pep:\
   :severe=3:log:units=units:delta=50:+jump=10:\
   :/* end  rpcc_timeout */:
rpcc_wait | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:\
   :!pep:severe=3:log:units=units:delta=0:+jump=10:\
   :/* end  rpcc_wait */:
rpcc_newcred | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:\
   :!pep:severe=3:log:units=units:delta=0:+jump=20:\
   :/* end  rpcc_newcred */:
rpcc_timers | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:\
   :!pep:severe=3:log:units=units:delta=0:+jump=20:\
   :/* end  rpcc_timers */:
rpcs_calls | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:\
   :!pep:severe=3:log:units=calls:delta=100:\
   :/* end  rpcs_calls */:
rpcs_badcalls | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:\
   :!pep:severe=3:log:units=bad calls:delta=20:+jump=10:\
   :/* end  rpcs_badcalls */:
rpcs_nullrecv | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:\
   :!pep:severe=3:log:units=units:delta=0:+jump=10:\
   :/* end  rpcs_nullrecv */:
rpcs_badlen | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:\
   :!pep:severe=3:log:units=bad_length:delta=30:+jump=10:\
   :/* end  rpcs_badlen */:
```

```
rpcs_xdrcall | :\
   :on:testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:\
   :!pep:severe=3:log:units=units:delta=100:\
   :/* end  rpcs_xdrcall */:
*********
  api
**********
# to use the API's, collect data yourself and write it to a file.
# Then edit one of the entries below, setting "file=" to the
# pathname of your file, "data" to the field number your data is
# in, and "label" to the field number an optional label is in. You
# can use one file for one or more api tests or use separate files
# for each one.
api1 | :\
   :off:file=:data=:label=:\
   :testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:!pep:\
   :severe=3:log:units=units:delta=0:\
   :/* end  api1 */:
api2 | :\
   :off:file=:data=:label=:\
   :testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:!pep:\
   :severe=3:log:units=units:delta=0:\
   :/* end  api2 */:
api3 | :\
   :off:file=:data=:label=:\
   :testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:!pep:\
   :severe=3:log:units=units:delta=0:\
   :/* end  api3 */:
api4 | :\
   :off:file=:data=:label=:\
   :testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:!pep:\
   :severe=3:log:units=units:delta=0:\
   :/* end  api4 */:
api5 | :\
   :off:file=:data=:label=:\
   :testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:!pep:\
   :severe=3:log:units=units:delta=0:\
   :/* end  api5 */:
api6 | :\
   :off:file=:data=:label=:\
   :testfreq=5:alarmfreq=60:mailer=/bin/mail:notify=root:!pep:\
   :severe=3:log:units=units:delta=0:\
   :/* end  api6 */:
```

```
*********
  inventory
**********
software | :\
   :on:testfreq=60:alarmfreq=1400:mailer=/bin/mail:notify=root:\
   :!pep:severe=3:!log:units=units:delta=0:\
   :/* end  software */:
# The hardware test only gets done when Events is started.
hardware | :\
   :on:mailer=/bin/mail:notify=root:!pep:severe=2:log:\
   :units=units:delta=0:\
   :/* end  hardware */:
```

Sample EventsCli Use

This section shows examples of using the Events Command Line Interface (EventsCli) to send alarms to PEP, SNMP, etc.

The first example:

```
EventsCli -c DatabaseA -T "high tps" -u "tps" -v 12345 -s  5 -t 501
```

means the DatabaseA is doing 12345 tps and has exceeded the high setting for the tps threshold. This event will be reported with the highest severity level (5). It is automatically reported to PEP and EMD, and an SNMP trap is sent using trap number 501.

The following example is a more general use of how to run EventsCli in a script.

```
#!/bin/sh
# This is one way the EventsCli might be used. This example program
# sends an alarm when it finds users that have been idle for more
# than one day.This script is OS specific, so you might have to
# make subtle changes for it to work on your system.
# This script could be called via crontab at what ever interval you
# deem appropriate, perhaps daily.
# This script is only an example, we don't claim it has any real
# usefulness nor is it warrented in any way.This is AS IS software.
#
TESTNAME="idle_user"
TESTVALUE=
UNITS=days
USERNAME=
SEVERITY=2 # we've arbitrarily chose this value.
TRAP=501   # any valid number will work, perhaps YOUR policy says
#            use 501 for this type of test.
export

# Begin Subroutines
#  Alarm is called recursively
Alarm()
{

  case $1 in
       "" ) exit ;;    # when there are no more arguments to process
         * ) USERNAME=$1 ;
             TESTVALUE=$2 ;
             echo EventsCli -n $TESTNAME -T $USERNAME -v $TESTVALUE
                            -u $UNITS -s $SEVERITY -t $TRAP ;
             shift ;
             shift ;
             Alarm $* ;;
  esac
}

# end of subroutines
#
# MAIN starts here
#get a list of users and how long they've been logged in
#Pass the list to the Alarm() subroutine
# Syntex is Alarm username idletime username idletime ...
Alarm `w | awk '{ printf "%s %s\n", $1, $4 }' |  grep $UNITS `
```