
Subject: Re: /dev/pty always gone
Posted by [Thomasd](#) on Wed, 16 Jan 2008 18:39:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

there you go:

```
# rpm -qf /lib/udev/devices/  
udev-095-14.9.el5
```

and

```
]# cat /sbin/start_udev  
#!/bin/sh  
#  
# start_udev  
#  
# script to initialize /dev by using udev.  
#  
# Copyright (C) 2004 Greg Kroah-Hartman <greg@kroah.com>  
#  
# Released under the GPL v2 only.  
#  
# This needs to be run at the earliest possible point in the boot  
# process.  
#  
# Based on the udev init.d script  
#  
# Thanks go out to the Gentoo developers for proving  
# that this is possible to do.  
#  
# Yes, it's very verbose, feel free to turn off all of the echo calls,  
# they were there to make me feel better that everything was working  
# properly during development...  
#  
# don't use udev if sysfs is not mounted.
```

```
sysfs_dir=/sys
```

```
export TZ=/etc/localtime
```

```
[ -d $sysfs_dir/class ] || exit 1  
[ -r /proc/mounts ] || exit 1  
[ -x /sbin/udev ] || exit 1  
[ -f /etc/udev/udev.conf ] && . /etc/udev/udev.conf  
udev_root=${udev_root-/dev}
```

```
. /etc/init.d/functions
```

```
prog=udev  
bin=/sbin/udev  
udevd=/sbin/udevd  
MAKEDEV="/sbin/MAKEDEV"
```

```
xargs_simple () {  
    if [ "$1" = "-n" ]; then  
        shift  
        MAXNR="$1"  
        shift  
    else  
        MAXNR=100  
    fi  
    NR=$MAXNR  
    ARGS=""  
    [ -z "$1" ] && set echo  
  
    while read line; do  
        if [ $NR -gt 0 ]; then  
            ARGS="$ARGS $line"  
            NR=$((NR - 1))  
        else  
            "$@" $ARGS  
            NR=$MAXNR  
            ARGS="$line"  
        fi  
    done  
    if [ -n "$ARGS" ]; then  
        "$@" $ARGS  
    fi  
}
```

```
make_extra_nodes () {  
    ln -snf /proc/self/fd $udev_root/fd  
    ln -snf /proc/self/fd/0 $udev_root/stdin  
    ln -snf /proc/self/fd/1 $udev_root/stdout  
    ln -snf /proc/self/fd/2 $udev_root/stderr  
    ln -snf /proc/kcore $udev_root/core  
  
    [ -d $udev_root/pts ] || mkdir -m 0755 $udev_root/pts  
    [ -d $udev_root/shm ] || mkdir -m 0755 $udev_root/shm  
    [ -a /dev/MAKEDEV ] || ln -s $MAKEDEV /dev/MAKEDEV;  
  
    if [ -x $MAKEDEV ]; then  
        for i in /etc/udev/makedev.d/*.nodes; do
```

```

        if [ -f "$i" ]; then
            cat "$i" | sed -e 's,#.*,g' | \
                xargs_simple -n 100 $MAKEDEV -x
        fi
    done
fi
for devdir in /etc/udev/devices /lib/udev/devices; do
    [ -d "$devdir" ] || continue
    pushd $devdir &> "$udev_root/null"
    set *
    if [ "$1" != "*" ]; then
        cp -ar "$@" $udev_root/
        pushd "$udev_root" &> "$udev_root/null"
        [ -x /sbin/restorecon ] && /sbin/restorecon "$@"
        popd &> "$udev_root/null"
    fi
    popd &> "$udev_root/null"
done
}

kill_udev() {
    if [ -x /sbin/pidof ]; then
        pid=`/sbin/pidof -x udevd`
        [ -n "$pid" ] && kill $pid
    fi
}

wait_for_queue() {
    local timeout=${1:-0}
    local ret=0
    if [ $timeout -gt 0 ]; then
        /sbin/udevsettle --timeout=$timeout
    else
        /sbin/udevsettle
    fi
    ret=$?
    if [ $ret -ne 0 ]; then
        echo -n "Wait timeout. Will continue in the background."
    fi
    return $ret;
}

export ACTION=add
prog=udev
ret=0
STRING="$Starting $prog: "
# propagate $udev_root from /sys

```

```
echo -n "$STRING"
```

```
# mount the tmpfs on ${udev_root%/}, if not already done
```

```
LANG=C awk \"$2 == \"${udev_root%/}\" && \"$3 == \"tmpfs\" { exit 1 }\" /proc/mounts && {  
    if LANG=C fgrep -q \"none ${udev_root%/}/pts \" /proc/mounts; then  
        PTSDIR=$(mktemp -d)  
        mount --move $udev_root/pts \"$PTSDIR"  
    fi  
    if LANG=C fgrep -q \"none ${udev_root%/}/shm \" /proc/mounts; then  
        SHMDIR=$(mktemp -d)  
        mount --move $udev_root/shm \"$SHMDIR"  
    fi  
    mount -n -o mode=0755 -t tmpfs none \"$udev_root"  
    mkdir -m 0755 $udev_root/pts  
    mkdir -m 0755 $udev_root/shm  
    if [ -n \"$PTSDIR\" ]; then  
        mount --move \"$PTSDIR\" $udev_root/pts  
        rmdir \"$PTSDIR"  
    fi  
    if [ -n \"$SHMDIR\" ]; then  
        mount --move \"$SHMDIR\" $udev_root/shm  
        rmdir \"$SHMDIR"  
    fi  
  
    ret=$((ret + $?)  
}
```

```
# returns OK if $1 contains $2
```

```
strstr() {  
    [ "${1#*$2*}" = \"$1\" ] && return 1  
    return 0  
}
```

```
getval() {  
    what=$1  
    shift  
    for arg; do  
        if strstr \"$arg\" \"$what=\"; then  
            val=${arg#${what}=*}  
            echo $val  
            return 0  
        fi  
    done  
    return 1  
}
```

```
make_extra_nodes
```

```
cmdline=$(cat /proc/cmdline)
```

```

kill_udevd > "$udev_root/null" 2>&1
if [ -f "/sys/class/tty/console/uevent" ]; then
    # trigger the sorted events
    echo -e '\000\000\000\000' > /proc/sys/kernel/hotplug
    if strstr "$cmdline" modprobedebug; then
        touch /dev/.modprobe_debug
    else
        rm -f /dev/.modprobe_debug
    fi
    /sbin/udevd -d
    ret=${ret + $?}
    if strstr "$cmdline" udevdebug; then
        /sbin/udevcontrol log_priority=debug
    fi
    /sbin/udevtrigger
    ret=${ret + $?}
    wait_for_queue $(getval udevtimeout $cmdline)
    ret=${ret + $?}
else
    echo -n " kernel too old for this udev version "
    /sbin/udevd -d
    ret=10
fi

ret=${ret + $?}
[ $ret -eq 0 ] && success "$STRING" || failure "$STRING"
echo
exit 0

```

so, here's what I found:

if I install from one of the CentOS 5 template (either full or minimal, same problem), it will work properly. I can restart the VM many times without problem.

When I run `vzyum <vm> update` it will trigger the problem and the pty will be missing each time I restart the VM.

Not sure if it matters but I have dag added to my yum repositories:

```

[dag]
name=Dag RPM Repository for Red Hat Enterprise Linux
baseurl=http://apt.sw.be/redhat/el5/en/i386/dag
gpgcheck=1
enabled=1

```