check-raid-status

```
#!/bin/sh

# raid_status - check the state of the RAID.

# This script works for various types of RAID devices.  (Currently, 3Ware,
# gmirror, BSd 'ar0' raids, zpool)
# WARNING: Install the proper CLI program for your 3ware card, if you use 3ware.

# Set up a cronjob like this:
# */16 * * * * /home/rudy/bin/raid_status CRON

### Copyright (c) 2006, Rudy Rucker All rights reserved.
### Redistribution and use of script, with or without modification, is
### permitted provided that the following condition is met:
###    Redistributions of source code must retain the above copyright
###    notice, this list of conditions and the following disclaimer.
### THIS SOFTWARE IS PROVIDED BY AUTHOR AND CONTRIBUTORS ``AS IS'' AND
### ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
### IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
### ARE DISCLAIMED.

# ----------- Change Log ------------
# Mon Oct 11 15:20:37 PDT 2004 - rudy
#  Original script.
# Tue Feb  7 01:28:07 PST 2006 - rudy
#  Added 9500 and 9550 support
# Fri Jun  9 10:38:33 PDT 2006 - rudy
#  works for 'ar' and 'tw' mirrored arrays
# Tue Sep 12 10:23:13 PDT 2006 - rudy
#  Added gmirror and realized that not all 3ware's are the same...
# Fri Jan 18 00:46:13 PST 2008 - rudy
#  going to add support for multiple geom mirrors. gm0s1, gm0s2, etc...
# Fri Jan 18 01:03:25 PST 2008 - rudy
#  added zpool status checking.  untested  :)
# Fri Jan 18 01:23:54 PST 2008
#  can check machines with multiple array types... got zfs and gmirror?
#  tested a machine with zfs and gmirror
# ---------------------------------------------------------------------

#
# Variables
#
TWCLI="/opt/3ware/CLI/tw_cli";
AWK="/usr/bin/awk";
GREP="/bin/grep";
VALID="OK";
CONTROLLER="c2";
UNITS="u0 u1";
PORTS="p0 p1 p2 p3 p4 p5";
hostname=$(hostname -f);
UNITSTATUS="";
PORTSTATUS="";

#
# Functions
#

function_get_ports ()
{
        echo;
}

function_get_units ()
{
        echo;
```

```
}

function_get_controllers ()
{
        echo;
}

function_report_unit_status ()
{
        TEMPUNITSTATUS=$1;
        if [ "$UNITSTATUS" = "$VALID" ];
        then
                echo "Unit-Status /"$CONTROLLER"/"$UNIT": Condition Good";
                $TWCLI info $CONTROLLER $UNIT;
                echo
        else
                # ERROR!  Either print to TTY or send an email, based on MODE (which is arg[1])
                if [ "$MODE" = "CRON" ]; then
                        $ESTATUS_CMD | $MAIL -s "[ERROR] Raid array on $HOST returned $STATUS" $EMAIL
                else
                        echo "ERROR condition"
                        $ESTATUS_CMD;
                fi
        fi
}

function_report_port_status ()
{
        TEMPUNITSTATUS=$1;
        if [ "$UNITSTATUS" = "$VALID" ];
        then
                echo "Condition Good";
                $TWCLI info $CONTROLLER $UNIT;
                echo
        else
                # ERROR!  Either print to TTY or send an email, based on MODE (which is arg[1])
                if [ "$MODE" = "CRON" ]; then
                        $ESTATUS_CMD | $MAIL -s "[ERROR] Raid array on $HOST returned $STATUS" $EMAIL
                else
                        echo "ERROR condition"
                        $ESTATUS_CMD;
                fi
        fi
}

function_check_unit ()
{

        UNITSTATUS=$(${TWCLI} /${CONTROLLER}/${UNIT} show | $GREP "^${UNIT} " | ${AWK} {'print $3'});
}

function_check_port ()
{
        PORTSTATUS=$(${TWCLI} /${CONTROLLER}/${PORT} show | $GREP "^${PORT}" | ${AWK} {'print $2'});
}

function_main ()
{
        for i in $UNITS;
        do
                UNIT=$i;
                echo "for UNIT: "$UNIT" Controller: "$CONTROLLER;
                function_check_unit;
                function_report_unit_status;
        done;

        for a in $PORTS;
        do
                PORT=$a;
                echo "for PORT: "$PORT" CONTROLLER: "$CONTROLLER;
```

```
                function_check_port;
                function_report_port_status;

        done;
}

function_main;
```