



Best Practices for Deploying Automation of NAS Health Check

EMC Proven® Professional Knowledge Sharing

Cheol Jang
Regional Software Specialist 2
EMC Global Services
Jang_cheol@emc.com

Table of Contents

Introduction.....	3
System Requirement.....	3
Health Check Flow	4
Sendmail Configuration of Linux PC	4
Health Check Script Review	7
autocall.sh	7
Celerra list and login information(/callhome/Serial_Number/data.list)	8
check.exp – interactive script	9
automail.sh	10
/callhome/Serial_Number/mail.list sample	12
E-Mail sample after problem detect.....	12
Conclusion	13
Future Plans	13

Disclaimer: The views, processes or methodologies published in this article are those of the authors. They do not necessarily reflect EMC Corporation's views, processes or methodologies.

Introduction

As we know, EMC's NAS Product is superb in performance. It is more stable and offers more strength than our competitors' products. The Auto Call-home Function monitors and detects errors and alerts corporate centers to take immediate action. However, despite this great functionality, sometimes it does not work properly due to a few fatal weaknesses.

For instance:

- System down due to ControlStation
- Problem cases not detected by Auto Call-Home
- Auto Call Home does not work due to change in customer's extension number
- Faulty registration of Auto Call-Home number caused by human error
- Cases of NAS system OS bugs preventing Auto Call-Home from working properly

If we can perform Health Check by conducting regular dial-ins to a NAS system, and send the result to the responsible account CE or system administrator by e-mail, it could increase system reliability dramatically by preventing mis-detected errors and reducing instances of issues mentioned above.

System Requirement

- PC (1PC can check 200 ~ 300 NAS system per day)
- Modem and Modem line
- NAS Version: 5.1.x and later version is recommended.
- Linux OS(In my case, Fedora is used)

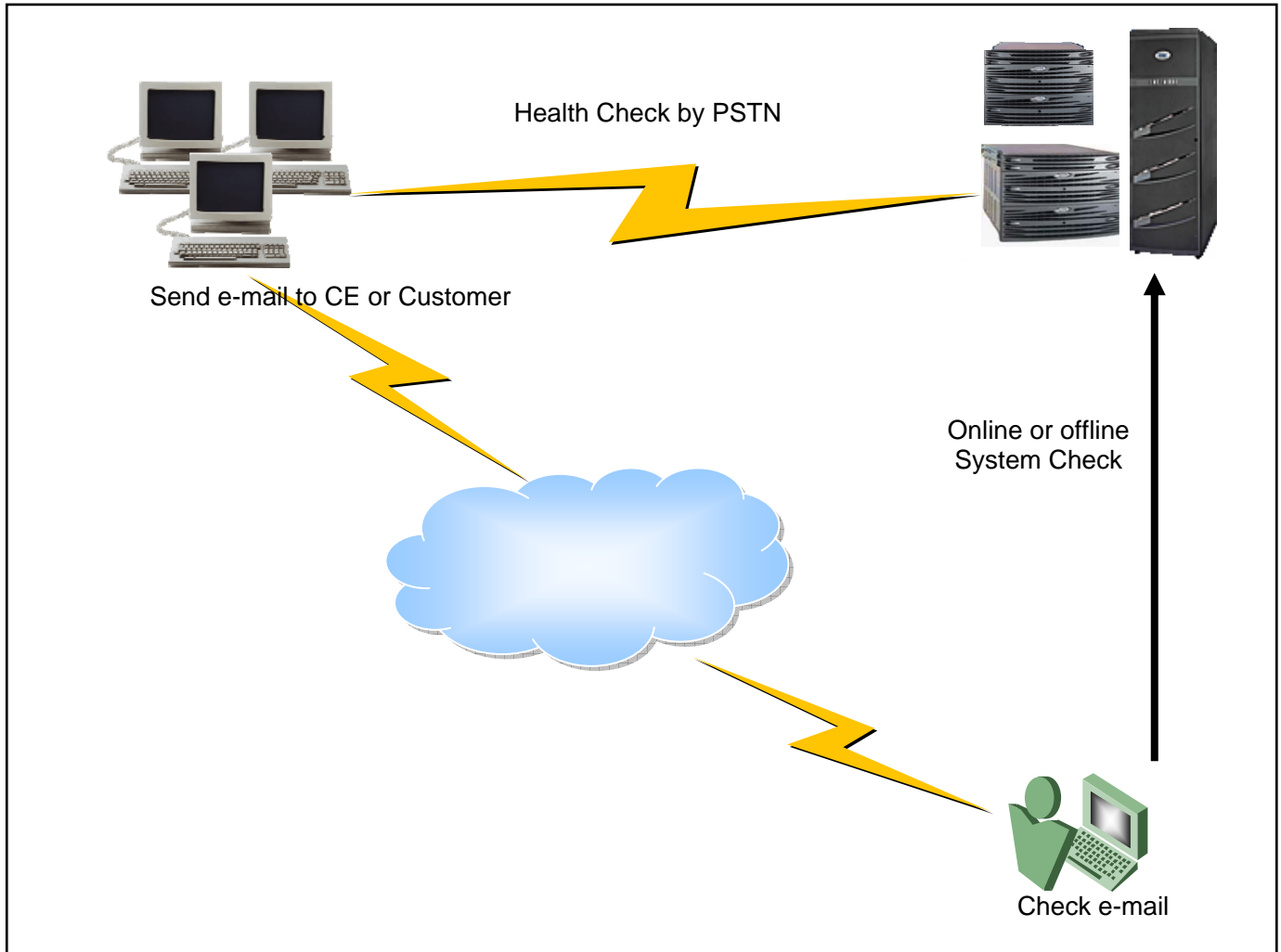
Any Linux should work but you need to install uucp, expect, sendmail and sendmail-cf package.

You can check like below.

```
-----  
[root@healthcheck1 /]# rpm -qa | egrep '(sendmail|uucp|expect)'  
expect-5.43.0-8  
sendmail-8.14.1-4.2.fc7  
sendmail-cf-8.14.1-4.2.fc7  
uucp-1.07-15.fc7  
[root@healthcheck1 /]#  
-----
```

Health Check Flow

NAS health Check System(Linux PC) → Call every NAS System → Log analysis in Linux PC
→ send e-mail to CE or Customer → online or offline system check → Fix the problem



Sendmail Configuration of Linux PC

Now you should setup your network environment and sendmail configuration.

This is my network configuration. You can refer to below.

```
[root@healthcheck1 /]# cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1    localhost.localdomain localhost
152.62.24.141 healthcheck1 healthcheck1.test.com
::1        localhost6.localdomain6 localhost6
```

```
[root@healthcheck1 /]# cat /etc/resolv.conf
nameserver 152.62.25.2
[root@healthcheck1 /]# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=healthcheck1
GATEWAY=152.62.24.1
[root@healthcheck1 /]#
```

If you want to send e-mail to EMC external, refer to below or primus emc151602.

```
[root@healthcheck1 /]# cat /etc/mail/sendmail.emc.mc
divert(-1)dnl
include(`/usr/share/sendmail-cf/m4/cf.m4')dnl
VERSIONID(`linux setup for Red Hat Linux')dnl
OSTYPE(`linux')dnl

dnl Allow sendmail to operate without DNS
define(`confSERVICE_SWITCH_FILE', `/etc/mail/service.switch')dnl
FEATURE(nocanonify)dnl
FEATURE(`accept_unqualified_senders')dnl

define(`SMART_HOST', include(`/etc/mail/smarthost'))dnl
define(`ALIAS_FILE', `/etc/aliases')dnl
FEATURE(`smrsh', `/usr/sbin/smrsh')dnl

dnl FEATURE(`always_add_domain')dnl

FEATURE(`allmasquerade')dnl
FEATURE(`masquerade_envelope')dnl
MASQUERADE_AS(include(`/etc/mail/masqueradedomain'))dnl

dnl Change "From" names through the genericstable
FEATURE(genericstable, `hash -o /etc/mail/genericstable')dnl
dnl Identify the domain to which the genericstable applies
FEATURE(`generics_entire_domain')dnl
```

```
FEATURE(`accept_unresolvable_domains')dnl
```

```
DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl
```

```
MAILER(smtp)dnl
```

```
[root@healthcheck1 /]#
```

```
[root@healthcheck2 mail]# cat /etc/mail/smarthost
```

```
mailhub.lss.emc.com
```

```
[root@healthcheck2 mail]# cat /etc/mail/masqueradedomain
```

```
emc.com
```

```
[root@healthcheck2 mail]# cat /etc/mail/genericstable
```

```
root Scotland_notifications
```

```
[root@healthcheck2 mail]# cat /etc/mail/service.switch
```

```
hosts dns files
```

```
aliases files
```

```
[root@healthcheck2 callhome]# head -2 /etc/passwd
```

```
root:x:0:0:NAS_HEALTH_CHECK:/root:/bin/bash
```

```
bin:x:1:1:bin:/bin:/sbin/nologin
```

```
[root@healthcheck2 callhome]#
```

Also, you must execute “m4 /etc/mail/sendmail.emc.mc > /etc/mail/sendmail.cf”.

Restart sendmail daemon like as “/sbin/service sendmail restart”.

Now type as below.

```
[root@healthcheck1 /]# mail -s test_message xxx@google.com
```

```
test is test.
```

```
.
```

```
Cc:
```

```
[root@healthcheck1 /]#
```

If you have the correct setup environment, you will receive this test message.

Health Check Script Review

There are two main scripts that start from cron table.

```
[root@healthcheck1 callhome]# crontab -l
00 10 * * * /callhome0/autocall.sh > /dev/null 2>&1
30 23 * * * /callhome0/automail.sh > /dev/null 2>&1
[root@healthcheck1 callhome]#
```

- a /callhome0/autocall.sh read modem number and password from /callhome-/Serial_Number/data.list, then call /callhome0/check.exp
- b /callhome0/check.exp check each NAS system.
- c /callhome0/autocall.sh logs to /callhome/Serial_Number/date.log.
- d /callhome0/automail.sh analyses /callhome/Serial_Number/date.log.
- e /callhome0/automail.sh reads mailing list from /callhome/Serial_Number/mail.list then sends e-mail to CE.

autocall.sh



Autocall.sh

```
[root@healthcheck1 callhome0]# cat autocall.sh
#!/bin/bash

SerialNumber=`ls /callhome` ← read Celerra List

for i in $SerialNumber
do
  sData=`cat /callhome/$i/data.list | awk '{print $2}'` ← read login data like password and modem
  number
  {
    j=0
    while true
    do
      /callhome0/check.exp $sData ← call check.exp script
      if [ $? -eq 0 ]
```

```

then
  break
fi
let j=j+1
if [ $j -ge 3 ] ← retry count
then
  break
fi
sleep 30
done
} > /callhome/${i} date +%m%d`.log 2>&1 ← write result of NAS health Check
sleep 30
done
[root@healthcheck1 callhome0]#

```

Celerra list and login information(/callhome/Serial_Number/data.list)

In my case, each Celerra has its own directory

```

[root@healthcheck1 /]# ls /callhome | head
APM00051304994_SKT_SUNGSU
APM00064605437_Supreme_Court_of_Korea_Seochon_HANMAE
CK200030505861_The_Segye_Times_HANMAE
CK200030505862_CHOSUN_UNIVERSITY_HOSPITAL_UNISYS
CK200030505878_SKC_SUNGSU
CK200030505881_Hallym_Scared_Heart_Hospital_HANMAE
CK200030800858_KT_IDC_Bundang_Nespot_HANMAE
CK200031600003_HANWHA_S_and_C_HANMAE
CK200032000405_ICU_DAEJEON_HANMAE
CK200032201062_SKT_SUNGSU
[root@healthcheck1 /]# cat /callhome/APM00051304994_SKT_SUNGSU/data.list
PhoneNumber: 9,3408-xxxx ← dial in number
hostname: ns500_celerra7 ← Control Station Host Name
passwd: sktxxxx ← login password
[root@healthcheck1 callhome0]#

```

check.exp – interactive script



Check.exp

```
root@healthcheck1 callhome0]# cat check.exp
#!/usr/bin/expect

set PhoneNumber [lindex $argv 0]
set hostname [lindex $argv 1]
set passwd [lindex $argv 2]
set timeout 60
set scripts /nas/tools/check_nas_upgrade
← If your Celerra is old system then you can't find /nas/tools/check_nas_upgrade,
← you can extract from NAS upgrd-ck script to /nas/tools directory.

spawn cu -l ttyS0 -s 9600
expect -exact "Connected."
send -- "ATZ^M\r"
expect -exact "OK"
send -- "ATZ^M\r"
expect -exact "OK"
set timeout 900
send -- "atdt $PhoneNumber\r"
expect {
    -exact "$hostname login:" { send -- "nasadmin\r" }
    -exact "NO CARRIER" { close ; exit 1 ; }
}
expect {
    -exact "Password:" { send -- "$passwd\r" }
    -exact "NO CARRIER" { close ; exit 1 ; }
}
expect {
    -exact "\[nasadmin@$hostname nasadmin\]\$" { send -- "su\r" }
    -exact "NO CARRIER" { close ; exit 1 ; }
}
```

```

expect {
    -exact "Password:" { send -- "$passwd\r" }
    -exact "NO CARRIER" { close ; exit 1 ; }
}
expect {
    -exact "\[root@$hostname nasadmin\]\#" { send -- "$scripts -pre\r" }
    -exact "NO CARRIER" { close ; exit 1 ; }
}
expect {
    -exact "\[root@$hostname nasadmin\]\#" { send -- "exit\r" }
    -exact "NO CARRIER" { close ; exit 1 ; }
}
expect {
    -exact "\[nasadmin@$hostname nasadmin\]\$" { send -- "rm -f
/nas/log/check_nas_upgrade.*.log\r" }
    -exact "NO CARRIER" { close ; exit 1 ; }
}
expect {
    -exact "\[nasadmin@$hostname nasadmin\]\$" { send -- "rm -f
/nbsnas/log/check_nas_upgrade.*.log\r" }
    -exact "NO CARRIER" { close ; exit 1 ; }
}
expect {
    -exact "\[nasadmin@$hostname nasadmin\]\$" { send -- "exit\r" }
    -exact "NO CARRIER" { close ; exit 1 ; }
}
[root@healthcheck1 callhome0]#

```

automail.sh



Automail.sh

You can ignore unimportant message in this script.

```

[root@healthcheck1 callhome0]# cat automail.sh
#!/bin/bash

```

```

SerialNumber=`find /callhome -maxdepth 1 -type d | grep -v callhome$ | cut -d / -f 3`
DATE=`date +%m%d`
for i in $SerialNumber
do
    modem=`strings /callhome/$i/$DATE.log | grep Checking | wc -l`
    CheckLog="/callhome/$i/$DATE.log"
    ModemProblem="/callhome0/modem_check.txt" ← Message file for modem or CS problem
    ← You can ignore unimportant message like as below.
    healthy=`strings /callhome/$i/$DATE.log | grep Checking | grep -v Pass | grep -i fail | grep -v "dart
release compatibility" | grep -v "resolv conf" | grep -v "cron jobs" | grep -v "FLARE is supported" |
grep -v "DM Hardware revision" | grep -v "Compat type file systems" | grep -v "NDU Integrated
software" | grep -v "Checking control lun size" | grep -v "Auto Assign setting" | grep -v "Backend
Storage Requirement" | grep -v "Checking if standby is down" | grep -v "hardware is supported" |
grep -v "Checking if auto assign are disabled for all luns" | grep -v "Checking unique id" | grep -v
"Checking boot files" | grep -v "Checking nas device map" | wc -l`
    if [ $modem -gt 30 ]
    then
        ModemStatus="$i Modem status is OK."
        if [ $healthy -eq 0 ]
        then
            HealthStatus="$i System status is OK."
            continue
        else
            sleep 5
            HealthStatus="$i has some problem, Please check the system."

            sTO=`cat /callhome/$i/mail.list | grep to: | awk '{print $2}'`
            sCC=`cat /callhome/$i/mail.list | grep cc: | awk '{print $2}'`
            /bin/mail -s "NAS $i has some problem" -c $sCC $sTO < $CheckLog
            sleep 5
        fi
    else
        sleep 5
        ModemStatus="$i Modem has some problem, Please check the modem line."
    fi
done

```

```

HealthStatus="$i System status is Not Available."
sTO=`cat /callhome/$i/mail.list | grep to: | awk '{print $2}'`
sCC=`cat /callhome/$i/mail.list | grep cc: | awk '{print $2}'`
/bin/mail -s "Can't access NAS $i" -c $sCC $sTO < $ModemProblem
sleep 5

fi
done
[root@healthcheck1 callhome0]#

```

```

[root@healthcheck1 callhome0]# cat modem_check.txt
Please check the modem or Control Station.
[root@healthcheck1 callhome0]#

```

/callhome/Serial_Number/mail.list sample

```

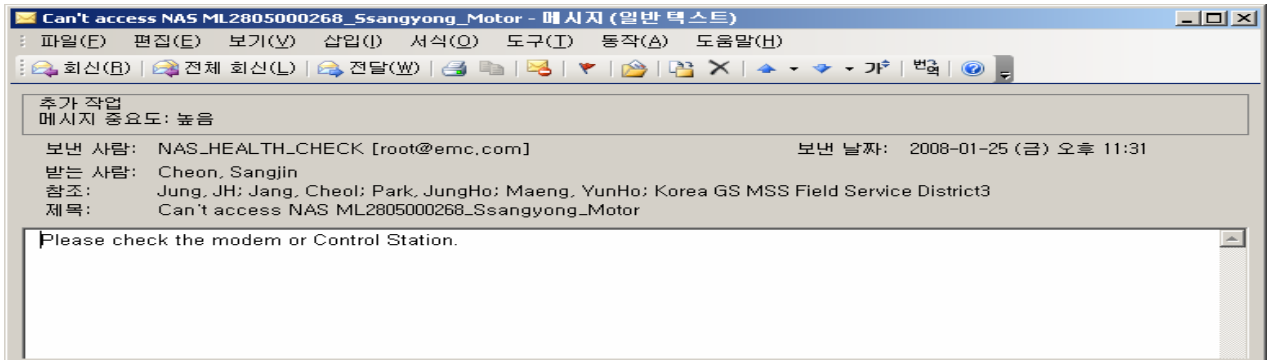
[root@healthcheck1 /]# cat /callhome/APM00051304994_SKT_SUNGSU/mail.list
to: kim_dogyun@emc.com
cc: choi_hankook@emc.com,jang_cheol@emc.com,maeng_yunho@emc.com,
_1cfe75@mail.corp.emc.com
[root@healthcheck1 /]#

```

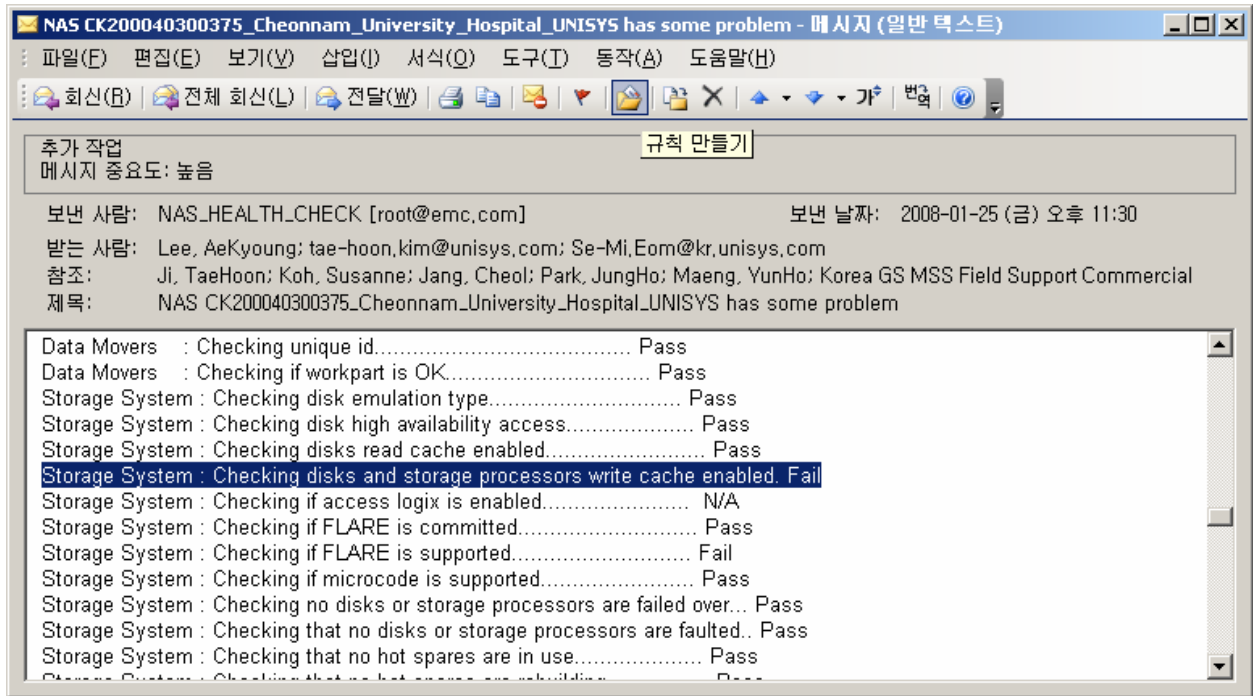
← _1cfe75@mail.corp.emc.com means mail group of “Korea GS MSS Field Service District2”

E-Mail sample after problem detect

- Modem or ControlStation problem



- CLARiON® write cache disabled



Conclusion

I set up the automated NAS health check system in Korea after conducting a simulation ten months ago. We have prevented more than 90 problem cases by taking corrective action prior to problem occurrence.

This document introduces the automated NAS health check system that I introduced in Korea. It also may be used as guide to reduce the EMC field engineers' or customer's system administrator workloads. As an added benefit, they will gain in-depth knowledge of NAS by studying the process.

Future Plans

At the moment, we provide this service free of charge to reduce engineer's workload and prevent site issues. When it becomes more stable, we will try to generate revenue by including it in one of the maintenance services.

Providing regularly scheduled 12 or 24 hour health checks to our premium customers would be a wonderful success story for EMC, and a great benefit to our customers.

Including this functionality in our implementation service to allow customers to check their NAS systems would be another attractive reason for customers to purchase EMC products.