SYSLOGD for OpenVMS[1] – Getting Started

Brett Cameron (brett.cameron@vmssoftware.com)
February 2018

These are instructions provide a brief overview to getting SYSLOGD running under on OpenVMS (8.4-1H1 or higher) with HP TCP/IP Services.

1. Create an account under which the SYSLOGD service will run, ensuring that the account has TMPMBX, SYSPRV, WORLD, and OPER privileges as per the example shown below (SYSPRV is needed to create a port less than 1024, and WORLD and OPER are needed to send broadcast messages). Note that these must be both authorized and default privileges.

   Be sure to create the directory for the account, and if you make the account captive or restricted make sure you create a LOGIN.COM.

```
Username: TCPIP$SYSLOG                      Owner:  SYSLOGD
Account:  SYSLOGD                           UIC:    [3655,30]
([TCPIP$AUX,TCPIP$SYSLOG])
CLI:      DCL                               Tables: DCLTABLES
Default:  SYS$SPECIFIC:[TCPIP$SYSLOG]
LGICMD:
Flags:
Primary days:   Mon Tue Wed Thu Fri
Secondary days:                   Sat Sun
No access restrictions
Expiration:            (none)    Pwdminimum:  6   Login Fails:     0
Pwdlifetime:         90 00:00    Pwdchange:      (pre-expired)
Last Login:            (none) (interactive),            (none) (non-
interactive)
Maxjobs:          0  Fillm:       128  Bytlm:       128000
Maxacctjobs:      0  Shrfillm:      0  Pbytlm:           0
Maxdetach:        0  BIOlm:       150  JTquota:       4096
Prclm:            8  DIOlm:       150  WSdef:         4096
Prio:             4  ASTlm:       300  WSquo:         8192
Queprio:          4  TQElm:       100  WSextent:     16384
CPU:        (none)   Enqlm:      4000  Pgflquo:     256000
Authorized Privileges:
  NETMBX       OPER         SYSPRV       TMPMBX       WORLD
Default Privileges:
  NETMBX       OPER         SYSPRV       TMPMBX       WORLD
```

2. Extract and copy the contents of supplied ZIP file into login directory for the TCPIP$SYSLOG account and set default to this directory.

3. Define the SYSLOG service (see SERVICE.COM):

```
$! Define and enable the syslog service (TCP/IP Services only)
$
$ tcpip set service syslog -
/proto=udp -
```

[1] An OpenVMS port of UNIX/Linux syslogd and logger. This is a slightly updated version of the code originally ported in 1995 by John Vottero, as found in the OpenVMS Freeware distribution.

```
/port=514 -
/flags=nolisten -
/inactive=0 -
/username=tcpip$syslog -
/process=syslogd -
/accept=(netw:127.0.0.0,netw:10.10.116.0) -
/file=sys$sysroot:[tcpip$syslog]startup.com -
/log=(all,file:sys$sysroot:[tcpip$syslog]syslogd.log)
$
$ tcpip enable service syslog
$ exit
```

Note that you will need to change (or omit) the `/ACCEPT` qualifier to specify your domain. You may also need to change the username, device, directory, and so on, depending upon how you have set things up on your system.

If you are happy with the contents of `SERVICE.COM`, then instead of manually entering the above command, simply run this procedure to set up the service:

```
$ @service.com
```

4. Create the command procedure to be run by the service (as specified by the `/FILE` qualifier in the service definition) or use the example script provided (`STARTUP.COM`). The contents of this file should include the following commands (as per `STARTUP.COM`) to define the configuration file and to run `SYSLOGD.EXE`.

```
$! Run the syslogd service
$!
$ define tcpip$syslog_config sys$login:syslogd.cfg
$ run sys$sysroot:[tcpip$syslog]syslogd.exe
$
$ exit
```

5. Enable the service (you may have done this already when creating the service, as per the example shown above or using the provided `SERVICE.COM`).

```
$ tcpip enable service syslog
```

Note that you will need to include this command in your system start-up (after starting TCP/IP Services) to ensure that the service is properly enabled whenever the system is rebooted.

6. Create a syslog configuration file or modify the sample file provided (`SYSLOGD.CFG`). The structure of this file is very similar to a UNIX syslog configuration file. Each line consists of one or more facility/severity combinations of the form <`facility`>.<`severity`> (you can use an "`*`" to signify all facilities or all severity's). If you specify more than one facility/severity separate them with commas.

After the facility/severity specification is at least one tab and then the destination for messages that match that facility/severity. The first character of the destination defines what type of destination it is in accordance with the following rules:

- `/` = Log to a file
- `@` = Forward to another node
- `%` = Send OPCOM message, % should be followed by a comma separated list of OPCOM classes

- Anything else is assumed to be a comma-separated list of usernames

The following illustrates a typical (simple) configuration file (lines beginning with "#" are treated as comments):

```
#
*.err           /sys$login:error.log
*.debug         /sys$login:debug.log
local0.*,user.* /sys$login:local.log
#
# Broadcast errors to JOHN and SYSTEM
#
# *.err JOHN,SYSTEM
#
# Send message to CENTRAL and TAPE operators via OPCOM
#
# local1.err    %CENTRAL,TAPES
```

Be sure to use tab's between facility/severity and destination names (do not use spaces).

7. Use the `LOGGER.EXE` utility program to send a message to `SYSLOGD` and verify that the service is operating correctly:

```
$ logger :== $sys$sysroot:[tcpip$syslog]logger.exe
$ logger "This is a test message"
```

You should see the `SYSLOGD` process start. If you encounter problems, look in the file specified in the `/LOG` qualifier when you defined the service.

8. You can also use the `LOGGER` utility to control the `SYSLOGD` process. The `-c` option can be used to send a command to `SYSLOGD`. Currently available commands are "`s`" and "`r`", where "`s`" means "shutdown" and "`r`" means "reopen the log files".

For example, the following command may be used to shut down the `SYSLOGD` process:

```
$ logger -c s
```