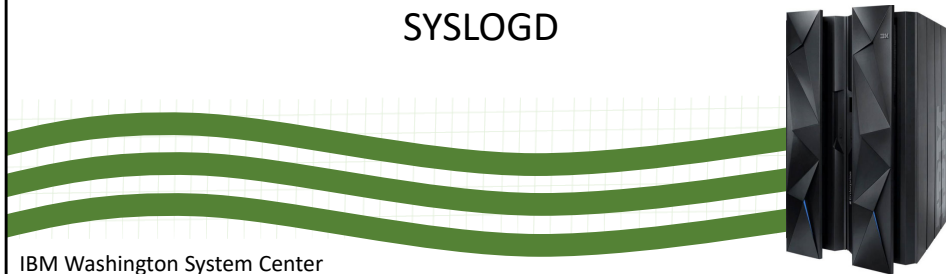


Securing and Encrypting Network Traffic with z/OS Communications Server and Policy Agent

Security Workshop

SYSLOGD



IBM Washington System Center
IBM Technical Sales Support

Trademarks

- **The following are Registered Trademarks of the International Business Machines Corporation in the United States and/or other countries.**
 - IBM
 - z/OS
- **The following are trademarks or registered trademarks of other companies.**
 - Microsoft is a registered trademark of Microsoft Corporation in the United States and other countries.
- All other products may be trademarks or registered trademarks of their respective companies.
- Refer to www.ibm.com/legal for further legal information.
- OSA-Express Features
- There are many different types of OSA-Express features. In this document where OSA is mentioned it refers to an OSA port on any of the OSA-Express features unless a specific OSA-Express feature is mentioned.

Agenda

- SYSLOG Daemon Overview
- Defining SYSLOG Daemon
- SYSLOGD Configuration File
- Automatic Archiving
- SyslogD ISPF Browser
- Log File Management
- Time Stamps
- Appendices:
 - Cron Daemon
 - Cron Example

SYSLOG Daemon Overview

Security Implementation can greatly increase the volume of log messages. You must warn your SYSLOG Daemon and MVS implementers about the increased logging activity.

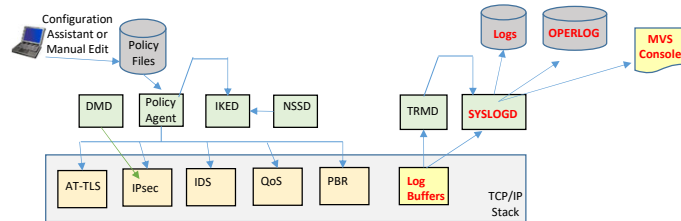


004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 4

Lots of Different Policy Types and Started Tasks



- Policy Agent
 - Installs Policies into the TCP/IP stack
- TCP/IP Stack
 - Enforces the Policies
- DMD (Defense Manager Daemon)
 - ipsec command can be used to install temporary IP Filter rules.
- IKED (Internet Key Exchange Daemon)
 - Required for IPsec Dynamic VPN (Virtual Private Network) Tunnels
- NSSD (Network Security Server Daemon)
 - Required for IKEv2
 - Provides central RACF certificate repository for remote IKED applications
 - Provides DataPower access to RACF
- TRMD (Traffic Regulation Management Daemon)
 - Required for log messages to syslogd for IP Filter, IPsec, and IDS
- SyslogD
 - **Recommended for logging**

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 5

Configuration Assistant can really help getting the whole environment setup.

What Happened to My Messages?



```
***** TOP OF DATA *****  
JES2 JOB LOG -- SYSTEM S 7 3  
--- TUESDAY, 20 JUL 1999 ----  
IEF695I START TCPIPIA WITH JOBNAME TCPIPIA IS  
$HASP373 TCPIPIA STARTED  
IEE252I MEMBER CTIEZB01 FOUND IN SYS1.PARMLIB  
EZZ0300I OPENED PROFILE FILE DD:PROFILE  
EZZ0309I PROFILE PROCESSING BEGINNING FOR DD:PROF  
.....  
EZZ0334I IP FORWARDING IS ENABLED  
.....
```

- SYSLOG Daemon (SyslogD) is the traditional Unix log repository.
- Prior to Unix becoming part of z/OS (MVS) all messages were written to the MVS System Log.
- Now that Unix is part of z/OS not all the messages are automatically sent to the MVS System Log, some Unix application messages are sent to syslogd instead.
- Rather than looking in multiple Unix application job logs it is recommended to use syslogd as a central log repository.

004_ZCS301_SYSLOGD

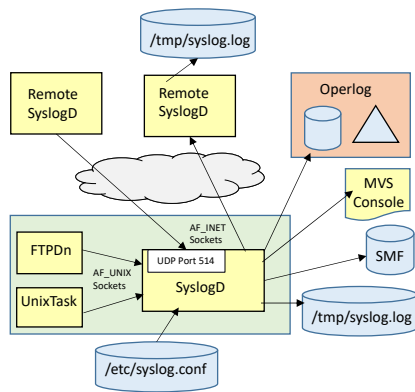
© Copyright IBM Corporation 2023

Page 6

In the olden days, many of the TCP/IP job messages appeared in the job log itself. Once Unix System Services (USS) -- formerly called "Open MVS" or "OMVS" -- was introduced, many of the job messages disappeared from the job log and began appearing in logs managed by UNIX: Syslog Daemon Logs.

SYSLOGD Environment

• One Single SYSLOGD

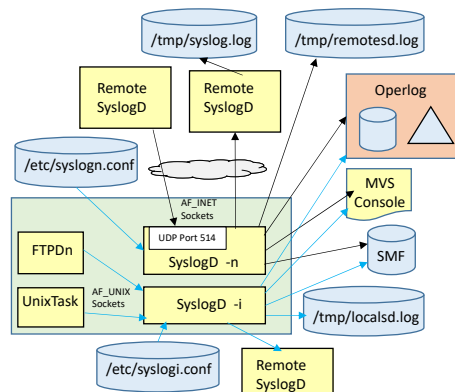


004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

• Or Two SYSLOGD

- One for Logging from remote systems
- One for Local Logging



Page 7

The syslog daemon reads and logs system messages to the MVS console, OPERLOG, log files, SMF, other machines, or users. If syslogd is not started, application data may appear on the MVS console.

You might have remote systems, critical UNIX servers, log data into your local log if there are critical messages that you need to see.

Syslog facility names are specified in the configuration file in /etc/syslog.conf with associated log data sets. (It is also possible to create the configuration file in an MVS dataset.) Depending on how the application has been coded, debug messages may go to one file and trace messages to another. Check each server documentation to see where they log debug messages, error messages, information messages and warning messages. Some will log all of them to the same syslogd destination.

AF_INET messages are received over the TCP/IP network. Internal Messages are received with AF_UNIX Sockets.

When syslogd is running, a file exists in /etc called /etc/syslog.pid. This file holds the process number of syslogd and can be used to stop syslogd or to request syslogd to reread its configuration file using a kill SIGHUP process number command. To open or create the syslog.pid file, the syslog daemon user ID must have a UID of 0.

SyslogD may be configured to forward messages to another SyslogD on another host instead of logging them to files, to send them directly to a specified user, to all users that are logged in on the system, to send messages to the MVS console, or to send messages to the Operlog.

At startup or when a SIGHUP signal is received SYSLOG Daemon reads its configuration file.

- UDP port 514 must be reserved to OMVS, and SyslogD should be started before or during TCP/IP startup. Otherwise some messages will flow to the MVS log and others will be lost.

OPERLOG:

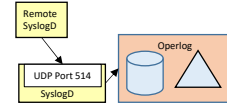
- If the destination for the remote messages is the MVS console or MVS system log designated with the /dev/console destination, consider whether logging these messages to the MVS operations log (that is, the OPERLOG log stream) -- designated by the /dev/operlog destination -- is a suitable alternative. Logging messages to the OPERLOG log stream is more efficient and consumes fewer system resources than logging messages to the MVS console. Operlog offloads the JES spool.
- The OPERLOG log stream must be configured and active before using the syslogd /dev/operlog destination. If the OPERLOG log stream is not active when syslogd attempts to log a message to the /dev/operlog destination, message FSUM1234 is logged with a facility.priority value of daemon.error. If the OPERLOG log stream later becomes active, message FSUM1235 is logged with a facility.priority value of daemon.info, and logging to OPERLOG automatically begins. For more information about using OPERLOG, see z/OS MVS Planning: Operations.

Two SYSLOG Daemons may run concurrently in z/OS:

- One can be used for receiving and sending log messages from and to the network. This syslog daemon is initiated with the "-n" parameter (stands for "network" processes)
- The other can be used for receiving and sending log messages for local processes. This syslog daemon is initiated with the "-i" parameter (stands for "internal" processes). The local syslogd can still send messages to remote Syslog Daemons; it just cannot receive messages.

Central Log Repository

- Receive Messages from Remote SyslogD systems
 - Central Monitor Location
 - Messages from Multiple SYSLOGD Systems Logged in One Place
 - Filter messages based on remote Source IP Address or Hostname
- Sysplex Operlog configured as log stream in the Coupling Facility
 - Centralizes messages for entire SYSPLEX
 - Contains z/OS generated messages and syslogd messages
 - Better performance than having SYSLOGD write to /dev/console
- Two instances of SYSLOGD
 - Improved SYSLOGD Performance
 - Logging from Local applications only mode (-i option)
 - Logging from remote Network SYSLOGD only mode (-n option)
 - Multi-threaded syslogd for improved performance and message capturing reliability
- Best Practice:
 - If you use both local and network logging, use two instances of syslogd
 - Remote messages do not interfere with local logging performance



004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 8

These enhancements are both for Security and for Performance in z/OS.

SYSLOGD log messages can now be collected from numerous network sources including Linux hosts and can be filtered to log to the desired destination based on the source IP address or hostname.

Log messages from network hosts can be written to the MVS operations log (operlog).

Operlog can be used in place of or in addition to MVS syslog (console log).

In a sysplex environment operlog can be configured as a log stream in the coupling facility.

- Provides a single sysplex-wide consolidated message log that contains z/OS generated messages and syslogd messages
- Better performance than writing to /dev/console
- Performance of syslogd is improved
- A local-only and a network-only instance may be run concurrently (i.e., TWO instances of Syslogd!)
- One instance in local only mode (-i option)
- One instance in network only mode (-n option)

If you use both local and network logging, IBM recommends that you use two instances of syslogd.

- Helps ensure that local syslogd logging is not adversely affected by the amount of remote messages being forwarded to z/OS. For messages arriving over the network, the rule can include the IP address or host name of the sender.

Defining SYSLOG Daemon



004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 9

SyslogD Startup

- SyslogD start command

```
syslogd [-c] [-d] [-D value] [-f conffile] [-F value] [-i] [-m markinterval] [-n]
        [-p logpath] [-u] [-x] [-?]
```

- c CREATE the logfiles and directories automatically
- d DEBUG mode
- D Default DIRECTORY permissions when created (by -c)
- f <config filename> specify configuration file
- F Default FILE permissions when created (by -c)
- i RECEIVE only AF_UNIX messages (not from network)
- m MINUTES between Mark Messages (used for syslogd testing)
- n RECEIVE only AF_INET messages from network
- p PATH (not recommended)
- u INCLUDE userid and jobname
- x Omits hostname lookup for messages received from remote SYSLOGD (performance improvement)

- Start SYSLOGD

- /etc/rc
- JCL Procedure and PROFILE.TCPIP AUTOLOG
- COMMNDxx member in PARMLIB

- Both SYSLOG.CONF and SYSLOG.LOG can be created with permission bits of 644.

- 6 = Owner can Read and Write
- 4 = Group can Read
- 4 = Other can Read

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 10

The permission bits for syslog.conf are usually set at 644 so that the owner can edit the file and others can browse the contents. A Superuser executes the file (SYSLOGD is associated with a superuser), so, in fact, any settings would work for the file. Originally the Unix System Services manuals were the source of information on configuring SYSLOGD. Now the z/OS Communications Server manuals have full information on configuring it.

The syslogd command recognizes the following options:

-c Create log files and directories automatically.

-d Run syslogd in debugging mode

-D Default access permissions (modes) to be used by syslogd when creating directories. If the -D option is not specified, the default value 0700 is used.

-f Configuration file name.

-F Default access permissions (modes) to be used by syslogd when creating log files. If the -F option is not specified, the default value 600 is used.

-i Do not receive messages from the IP network. This option is mutually exclusive with the -n option. If syslogd is started with the -i option, another instance of syslogd can be started with the -n option.

-m Number of minutes between mark messages. The default value is 20 minutes.

-n Receive messages from only the IP network. This option is mutually exclusive with the -i option. If syslogd is started with the -n option, another instance of syslogd can be started with the -i option.

-p Path name of z/OS UNIX character device for the datagram socket. The default value is /dev/log. Note: This option is not necessary unless you have changed the default location of the UNIX datagram socket. It is not generally useful. If you selected the -p option, syslogd will not function properly if you do not identify Unix datagram socket correctly.

-u For records received over the AF_UNIX socket (most messages generated on the local system), include the user ID and job name in the record.

-x Disable host name resolution for messages received from the IP network. This option is mutually exclusive with the -i option. Using this option can improve the performance of syslogd when processing messages received from the IP network. It has no effect for local messages. When you use this option, the IP address (instead of the host name) of the origin host is logged, along with the message text. If the host name can be determined from the rule without having to make a resolver call, the host name is used instead of the IP address. When the -x option is not used, syslogd always attempts to resolve the host name associated with a log message arriving from the IP network. If the host name cannot be determined, the IP address is logged as the message origin instead of the host name.

-? Show syslogd command-line options.

Unix Permission Bits for unix Files

File Owner UID	File Owner GID	Ex- tended Attri- bute	Set UID	Set GID	Sticky	Owner			Group			Other			File Owner	Audit
						Read	Write	Exec- ute	Read	Write	Exec- ute	Read	Write	Exec- ute		
						2^2 (4)	2^1 (2)	2^0 (1)	2^2 (4)	2^1 (2)	2^0 (1)	2^2 (4)	2^1 (2)	2^0 (1)		
						↓	↓	↓	↓	↓	↓	↓	↓	↓		
						Permission of 755 is:	1	1	1	1	0	1	1	0	1	
						Permission of 644 is:	1	1	0	1	0	0	1	0	0	

For Log Files and Configuration Files, Permission bits of 644 are usually adequate.
For directories, Permission bits of 755 are usually adequate, unless a user needs to write to a directory he does not own.

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 11

Extended Attributes is an extension to UNIX by z/OS Unix System Services. You can view these attributes with the command "ls -E." You can manipulate the settings of the attributes if you are authorized to run the "extrattr" command under UNIX. Extended attributes in z/OS refer to the ability to run executable files in APF authorized mode or as program-controlled in the unix (versus program-controlled from an MVS library).

Permission of 755:

- Owner can Read, Write and Execute
- Group can Read and Execute
- Anyone else can Read and Execute

Permission of 644:

- Owner can Read and Write
- Group can Read
- Anyone else can Read

/etc/rc

```
# Start the SYSLOG daemon for logging UNIX activity
_BPX_JOBNAME='NM2ASYSL' /usr/sbin/syslogd -f /etc/syslog.conf &
# /usr/sbin/syslogd -f /etc/syslog.conf &
.....
sleep 5
echo /etc/rc script executed, `date`
```

- Jobname "NM2ASYSL" if "_BPX_JOBNAME='NM2ASYSL'"
- Jobname "SYSLOGDn" if "_BPX_JOBNAME='SYSLOGD'"
- Jobname "ETCRCn" if started without "_BPX_JOBNAME="

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 12

SYSLOGD can be started only by a Superuser. If it is started with JCL, then the associated USERID for the STARTED task name must be a Superuser. If SYSLOGD is started via the /etc/rc file, then automatically the originally defined SUPERUSER will start it. The sample "rc" file issues "export _BPX_JOBNAME='ETCRC'," but you can override this jobname for specific daemons started up in /etc/rc if you use _BPX_JOBNAME= in the body of the /etc/rc configuration file.

The following notes are reminders about requirements for Superuser definitions for TCP/IP in general. (NOTE: SYSLOGD's user does not need to be authorized for BPX.DAEMON.)

Define a superuser (UID=0 user) and group for the TCP/IP started task and assign it to the TCP/IP procedure name:

- ADDGROUP OMVSGRP OMVS(GID(1))
- ADDUSER TCPIP DFLTGRP(OMVSGRP) OMVS(UID(0) HOME('/') PROGRAM('/bin/sh'))
- RDEFINE STARTED TCPPROC.* STDATA(USER(TCPIP) GROUP(OMVSGRP))

The RouteD and OMROUTE server and the SNMP agent also need superuser authority. You can reuse the TCPIP user ID or create separate started task user IDs with UID=0

- RDEFINE STARTED OROUTED.* STDATA(USER(TCPIP) GROUP(OMVSGRP))
- RDEFINE STARTED OSNMPD.* STDATA(USER(TCPIP) GROUP(OMVSGRP))

The FTP server and more of the servers that are started via INETD, need superuser authority and optionally Daemon authority. Define a separate user ID with UID=0 and READ authority to the BPX.DAEMON facility class. All UNIX-based server functions that need to be able to change user security environment based on an entered user ID and password, need READ authority to BPX.DAEMON.

- ADDUSER TCIPSRV DFLTGRP(OMVSGRP) OMVS(UID(0) HOME('/') PROGRAM('/bin/sh'))
- RDEFINE STARTED FTPD.* STDATA(USER(TCIPSRV) GROUP(OMVSGRP))

Common errors at startup of SYSLOGD and CRON...

When starting syslogd and you receive the following error:

BPXF024I (IBMUUSER) Aug 11 06:18:45 syslogd: cannot create /dev/log:

- EDC8114I Address family not supported.
- This means that AF_UNIX has not been defined in BPXPRMxx member; adding this will require an IPL.
 - FILESTYPE TYPE(IBMUDS) ENTRYPOINT(BPXTUINT)
 - NETWORK DOMAINNAME(AF_UNIX)
 - DOMAINNUMBER(1)
 - MAXSOCKETS(2000)
 - TYPE(IBMUDS)

JCL Procedure

Proc

```
//NM2ASYSL PROC MODULE='SYSLOGD',
//          PARM='-f /etc/syslog.conf'
// *
// *
//SYSLOGD EXEC
PGM=&MODULE,REGION=4096K,TIME=NOLIMIT,
//          PARM='POSIX(ON) ALL31(ON) /&PARMS'
//SYSPRINT DD SYSOUT=*
```

PROFILE.TCPIP

```
AUTOLOG 5
NM2ASYSL          ; SYSLOG Daemon PROC
ENDAUTOLOG

PORT
514 UDP OMVS      ; SYSLOG Daemon
```

/etc/services

```
syslog          514/udp
```

Start

```
S proc_name
ie. S SYSLOGD
```

Display

```
D A,SYSLOGD*
IEE115I ...
JOBS  M/S  TS USERS  SYSAS ...
00004 00011 00001 00035 ...
SYSLOGD1 STEP1  OMVSKERN ...
```

Stop

```
P proc_name
ie. P SYSLOGD1
```

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 13

Sample JCL procedure is SEZAINST(SYSLOGD).

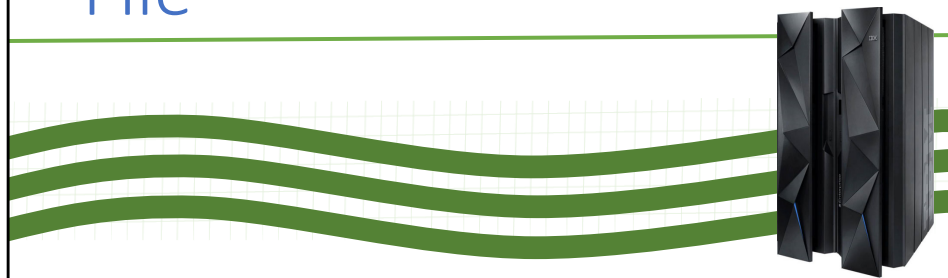
You may start the SYSLOG Daemon with various options:

- f /etc/syslog.conf
 - Identifies the name and location of the SYSLOGD configuration file
 - This option is necessary only if the path and name of the SYSLOGD configuration file is anything other than the default of "/etc/syslog.conf."

SYSLOGD must be started prior to any procedure that is to write its messages to the SYSLOGD log file. If there is no TCP/IP transport active when syslogd starts or if TCP/IP is recycled, syslogd will establish or reestablish communication with TCP/IP when it becomes available.

If you intend to receive log data from or send log data to remote syslogd servers, you must place the syslog service in the /etc/services file or services dataset.

SYSLOGD Configuration File



004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 14

SYSLOGD Configuration File

- SYSLOGD configuration file contains two types of statements:
 - Logging Rules
 - Archive Configuration Statements
- Sample is located in the TCP/IP unix sample directory:
 - /usr/lpp/tcpip/samples/syslog.conf
- If you update syslog.conf, you can request SYSLOGD to re-read the configuration file without restarting SYSLOGD by sending a SIGHUP signal:
 - Modify syslogd
 - /F SYSLOGD,RESTART
 - kill -SIGHUP <pid number of syslogd>
 - kill -1 <pid> where -1 is the number 1

Logging Rules

- Logging rules in /etc/syslog.conf
- Each logging rule has an Identifier and a Destination
- SYSLOGD startup without -u

- Identifier consists of two parameters

IDENTIFIER	DESTINATION
Facility_Name.Priority_Code	destination_path
ie.	
IDENTIFIER	DESTINATION
daemon.err	/tmplog/daemon.errlog

- SYSLOGD startup with -u

- Identifier consists of four parameters

IDENTIFIER	DESTINATION
User_ID.Job_Name.Facility_Name.Priority_Code	destination_path
ie.	
IDENTIFIER	DESTINATION
user9.ftpd1.daemon.err	/tmplog/daemon.errlog

Both facility name and priority code are predefined. (That is, you cannot establish new facility names or priority codes.)

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 16

This is a sample of how an application might invoke the writing of messages to SYSLOGD.

Sample C-source program that uses SyslogD:

```
#include <syslog.h>
openlog("oec", LOG_PID, LOG_LOCAL0);
syslog(LOG_INFO, "Hello from oec");
closelog();
```

The above sample resulted in the following SyslogD output:

```
May 26 11:27:51 mvs18oe oec[3014660]: Hello from oec
```

The default syslogd configuration file is /etc/syslog.conf. This file can also reside in an MVS PDS.

It may be overridden during start of syslogd using a startup option:

```
/usr/sbin/syslogd -f /u/sysprog/mysyslogd.conf
```

The syslog.conf file contains rules by which syslogd determines what to do with log messages it receives from programs in this system or from other systems.

A logging rule consists of an identifier and a destination.

The identifier consists of at least two fields: The facility name and the priority code.

daemon is one of the standard, traditional facility names.

err is one of the standard, traditional priority codes.

All programs, also user-written server programs, may use the syslogd server. The API to use syslogd consists of three functions:

```
openlog      to open a logging channel for a specified facility name
syslog       to send messages with a specified priority code
closelog     close the logging channel
```

A program that uses these functions must include syslog.h

The logging channel is an AF_UNIX socket communication (/dev/log).

Facility Name

kern	Unix kernel messages
user	Default facility used when no other category applies
Mail	Mail system messages
news	Usenet system messages
uucp	uucp messages
daemon	Server messages
auth	Authorization messages
authpriv	Same as auth
cron	cron system messages
lpr	Printing system messages
local0-7	Facility names meant for local use TelnetD uses local1 to log its messages IKE uses local4
*	Placeholder used to represent any facility name
mark	Provides heartbeat messages

Application usage of Facilities is documented in a table in the Syslog daemon chapter of the IP Config Ref manual.

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 17

Logging rules are maintained in `/etc/syslog.conf`.

Each rule consists of an identifier and a destination.

The identifier consists of a facility name and a priority code.

syslog facilities: user, mail, news, uucp, daemon, auth (authpriv), cron, lpr, local0-local7, mark

mark is special; it means that syslogd should log heartbeat messages to the specified destination; priority must be info or higher for this to work.

mark.info `/var/log/messages`

Priority Codes

emerg	Emergency - system is becoming unusable
panic	Same as emerg
alert	Immediate action is required
crit	Critical condition - device or is becoming unusable
error	Error condition
warning	Warning condition
notice	Normal, but significant condition
info	Information message
debug	Debugging message
none	Placeholder used to represent none of the priorities
*	Placeholder used to represent all priority codes

- NOTE: A priority code includes all above priorities.
 - Emerg is the highest priority.

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 18

SyslogD has nothing to do with the standard MVS SYSLOG function, except from the following two points:

- If SyslogD is not started when an application tries to send it a message to log, the message may be sent (depends on options on the openlog() call) to the MVS SYSLOG (the MVS console).

- SyslogD can be configured to send critical messages to the MVS SYSLOG (the MVS console).

syslog priorities, in order from highest to lowest: emerg (panic), alert, crit, err (error), warning (warn), notice, info, debug, none. All priorities \geq the specified priority will be logged.

daemon.info logs all messages with priority info, notice, warning, err, crit, alert, or emerg if the facility is daemon.

none is special. It means that no messages from the specified facility are matched:

*.err;daemon.none /var/log/errors

Destination

- A file in the hierarchical file system
auth.* /tmplog/syslogd/auth.log
- One or more local shell users
facility_name.priority_code user1,user2
facility_name.priority_code *
- A SyslogD server on another host
facility_name.priority_code @myaixserver
- Remote SYSLOGD messages can be separated by remote IP Address or Hostname
(host14.xyz.com).*. * /var/log/rslog14.log
(10.42.15.0/24).*. * /var/log/rslog15.log
- MVS Operlog (if it has been implemented)
*. * /dev/operlog
- The MVS console
facility_name.priority_code /dev/console
- Don't log the messages with Priority Code of ".none":
mail.err /var/log/mail.log
*.err;mail.none; /var/log/err.log

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 19

The most traditional setup is to direct log messages to various files in the directory you have set aside for log files and trace output. Typically you route messages to individual files based on the facility name, and maybe a separate file for the most critical messages.

The usual rule of thumb is to start SYSLOG Daemon prior to TCP/IP stack startup; otherwise messages are lost. Although you should still follow this rule of thumb, in fact, if you were to direct some messages to the MVS console with "/dev/console," you may notice that a recycle of the SYSLOG daemon while TCP/IP is up and running does NOT cause those messages to be lost. Note the use of *.none in the last example.

- none is special; it means that no messages from the specified facility are matched. This will prevent double logging.

RECOMMENDED: Exploit OPERLOG for SYSLOGD IO. This offloads the spool files and provides better IO performance.

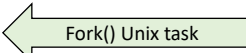
When Remote SYSLOGD messages are received the UserID and JobName are not known. The IP Address or Hostname of the Remote SYSLOGD may be used to separate the messages instead. Since the UserID and JobName are not known there is no need for this support for 4 parameter Identifiers, only the facility_name and priority_code are supported. It is possible to create rules that contain an IP Address or Hostname in one rule along with userid.jobname.facility.priority:

```
(host3.abc.com).daemon.*;user3.ftpd*.daemon.* /var/log/daemon.log
```

SYSLOGD -u and Symbols

- When SYSLOGD is started with -u the Identifier consists of four parameters

- IDENTIFIER	DESTINATION
- User_ID.Job_Name.Facility_Name.Priority_Code	destination_path
- User_ID and Job_Name enables separation of logged messages

- *.ftpd*. *.*	/var/log/ftpd.log	
- user7.*.*	/var/log/user7.log	
- user8.ftpserv1.* *	/var/log/user8_ftpserv1.log	
- user9.ftpserv2.* *	/var/log/user9_ftpserv2.log	
- File names can be created using symbols for date:
 - userx.job3.*.* /var/%Y/%m/%d/userx_job3.log
 - %Y represents the current year
 - %m represents the current month
 - %d represents the current day
 - Path is created for symbols even without SYSLOGD startup with -c

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 20

Identifiers User_ID and Job_Name is useful for separating logging of LOCAL SyslogD messages.

Syslog Daemon in z/OS has many enhanced capabilities over SYSLOGD on other platforms:

- User ID and jobname passed onto local SyslogD by AF_UNIX sockets
- User ID and jobname can be used in logging rules as selection criteria and can be included in logged messages
- Ability to turn off receiving messages from remote SyslogD servers but still be able to send to other SyslogD servers
- Ability to run TWO SyslogD servers (network and local)
- Ability to allow SyslogD to create new directories and log files

Symbol format strings supported - whatever standard strftime() function supports.

Example:

%d	two-digit day of month (01-31)
%m	two-digit month of year (01-12)
%Y	four-digit year (2000-)

Use %% to represent a % - migration issue.

If the log file name contains date symbols format, syslogd will create it if it doesn't already exist.

Any file name without symbols, syslogd won't create files or directories unless...

- With the -c start option, syslogd will attempt to create ANY log file and/or directory that doesn't already exist.

See Language Environment C/C++ Run-time Library Reference for the complete set of strftime() format strings supported on z/OS. Or use the simple ones mentioned here.

If you choose to manage SYSLOGD log files with Cron...

Use CRON job to have SyslogD create new files at midnight:

```
0 0 * * * kill -HUP `cat /etc/syslog.pid`
```

Use another CRON job to remove old log files:

```
0 1 * * * /usr/local/bin/rmoldlogs
```

The log files that are defined in /etc/syslog.conf need to exist before starting syslogd unless you have initialized SYSLOGD with the "-c" parameter (for "create").

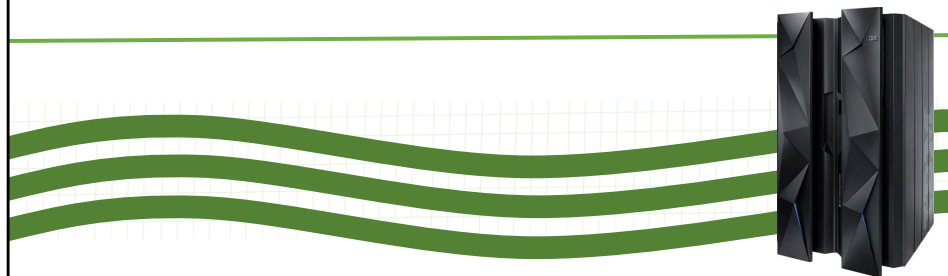
If they do not exist, the message 'No such file or directory exists.' will be displayed. To create the files issue the following command:

- touch /tmp/log.filename or touch /tmplog/log.filename

Permission bits of 644 are adequate for log files, since most processes that write to the log files are associated with a superuserid (UID=0).

Common problems with CRON are the failure to set up the CRON QUEUEDEFS.

Automatic Archiving



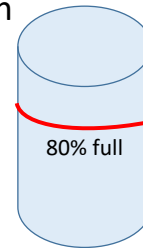
004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 21

Archive Configuration Statements

- Also referred to as Global syslogd Configuration Statements
- Specified in SYSLOGD configuration file
- ArchiveThreshold
 - Define Archive due to the file system being a percentage in use (full)
- ArchiveTimeOfDay
 - Define Archive due to a time of day
- BeginArchiveParms/EndArchiveParms
 - Define a prefix to use for Archiving



Archive when Threshold is reached, or by Time of Day, or both.

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 22

There are four different Archive Statements:

ArchiveCheckInterval *minutes*

Interval that syslogd checks file system utilization (default 10 minutes)

ArchiveThreshold *percentage*

Percentage of file system use that triggers archive (default 70)

0 prevents archive due to threshold

ArchiveTimeOfDay *time*

Local time of day to perform archive (no default)

BeginArchiveParms/EndArchiveParms

BeginArchiveParms

DSNPrefix *prefix* Unit *unit* Volume *volume* MgmtClas *class* StorClas *class* RetPd *days*

prefix specifies the archive data set name prefix value

All other parameters are optional and define information for the allocated data set: unit, volume, management class, storage class, and retention period.

Archive Destination

- IDENTIFIER DESTINATION -F *file_acc_per* -D *dir_acc_per* -N *archive_prefix* -X
 - *file_acc_per* defines the unix permissions for allocated file
 - *dir_acc_per* defines the unix permissions for allocated directory
 - *archive_prefix* is used with the BeginArchiveParms statement
 - -X causes the log file to be deleted and a new file created
- Archive sequential data set is created with name:
 - prefix.archive_prefix.data_suffix.time_suffix
 - prefix is defined on BeginArchiveParms
 - archive_prefix is defined by -N on Logging Rule
 - data_suffix is the date Dyyymmdd
 - time_suffix is the time Thhmmss
- When Archive is Generation Data Group (GDG) data set:
 - prefix.archive_prefix.gdg_suffix
 - gdg_suffix is an automatic unique value, ie. G0007V00

004_ZCS301_SYSLOGD

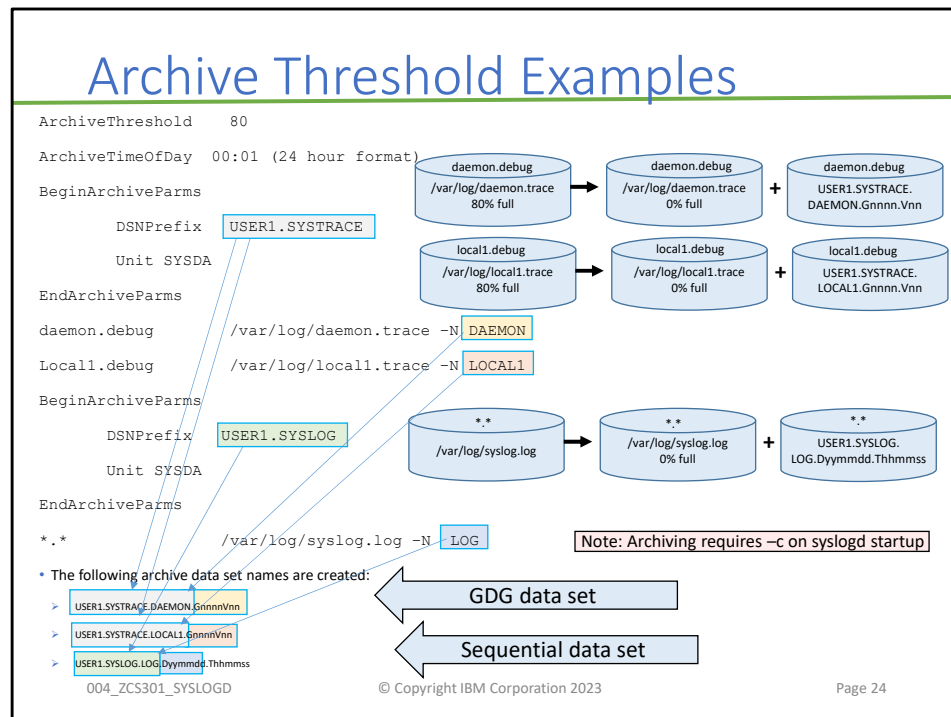
© Copyright IBM Corporation 2023

Page 23

In addition to Identifier and Destination:

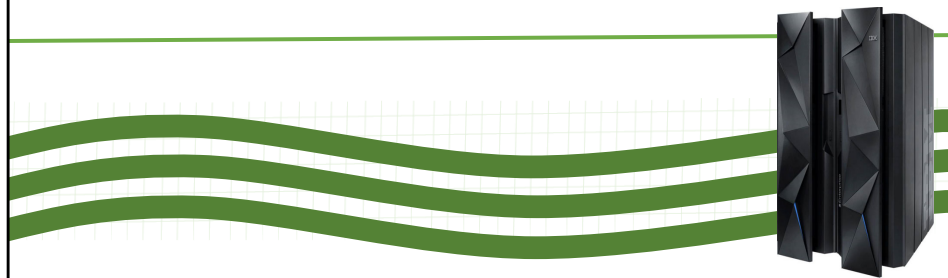
-F
-D
-N
-X

The Archive data set name is different depending on if the data set is a sequential data set or GDG data set.



When Automatic Archiving is triggered (by Threshold Percentage or Time) the log file is copied into an Archive file, leaving the active log file empty for more logging.
 A Catalog Lookup is done. If a GDG base is listed then a Generation data set is created. Otherwise a Sequential data set is created.

SyslogD ISPF Browser

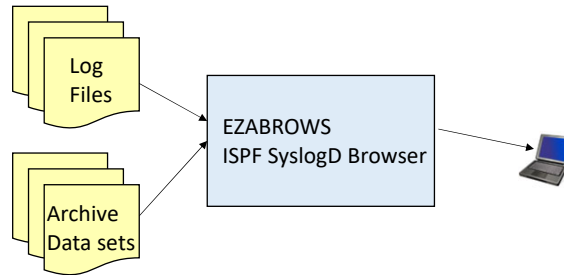


004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 25

ISPF SyslogD Browser



- **EZABROWS**

- Reads SYSLOGD configuration file to learn active log files and archive data sets
- Displays and Searches active log files and archive data sets

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 26

New in OS/390 V1R11.

See the IP Configuration Reference for directions how to define library access to TSO Logon procedure, CLIST, or REXX exec. The manual also details how to add the ISPF syslogd Browser to the ISPF primary option.

The syslogd browser provides an easy-to-use interface to access the messages syslogd has captured on a z/OS system.

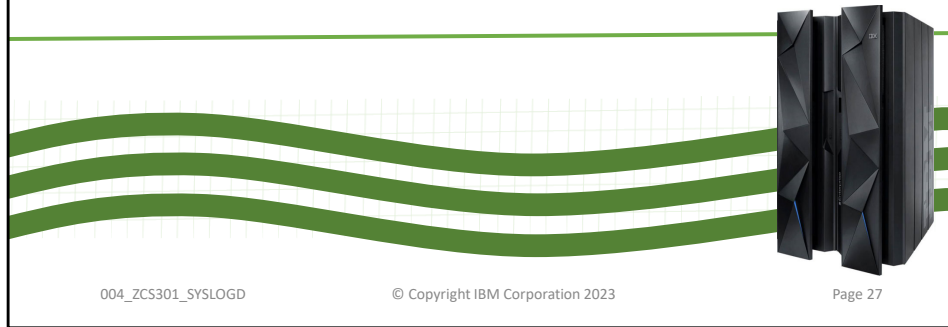
The browser supports accessing archived syslogd messages, when such archives are created using the syslogd archive function (archive to MVS data sets: GDG data sets or sequential data sets with date/time info in the LLQs). Archives based on %-symbols are also supported as long as such archive files remain in the directory in which they were originally created.

The browser provides a search mechanism that allows you to search selected active z/OS UNIX files and or archives based on various search arguments.

The browser also provides a 'guide-me' function that allows you to enter syslogd rule criteria and guide you to the active z/OS UNIX file or files such messages go to.

The browser exploits the syslogd archive functions if they are used, but the browser does not depend on use of those functions.

Log File Management

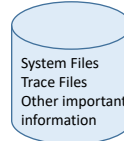


004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 27

SyslogD File Location



Root unix at Mountpoint "/"
OMVS.HFS.TMP at Mountpoint "/tmp"

- Root unix Mountpoint "/"
 - Do not log to files that fill up the MVS dataset that is mounted at Root "/".
 - Root "/" is needed for System Files, Trace Files, and other important files.
- Mountpoint "/tmp"
 - Do not log to files that fill up the MVS dataset that is mounted at "/tmp".
 - "/tmp" may be needed for log data that cannot be rerouted to another path.
- Temporary Files System (TFS)
 - Do not use a TFS syslogd logging because you will lose important messages if the system crashes.
- Mountpoint "/var"
 - If you use "/var" for syslogd log files mount an MVS dataset at that mountpoint to avoid interfering with other paths.
- Mountpoint "/tmplog"
 - Create a directory like "/tmplog" and mount an MVS dataset at that mountpoint for syslogd log files.
- No matter what directory path you use, make sure you monitor it to be sure syslogd logging is not interrupted due to lack of space.
 - df (shows all usage)
 - df -P /tmp (shows usage for mount point /tmp)

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 28

At the very least you should keep your customization file (usually at /etc) and your logging file (usually /tmp) separate -- that is, they should have separate mountpoints.

Note that you may decide to direct your logging to an installation-defined mountpoint, for example: /tmplog.

Log files can grow very large. Usually you will want to keep them isolated from other unix files by defining them in files that are defined in a directory (usually /tmp) that is mounted separately.

IMPORTANT NOTE: Although the usual location for logging files is "/tmp," you may want to mount on an entirely different directory. This would protect your ability to execute certain system functions that rely on the availability of "/tmp" space.

"/tmp" contains not only the files you direct to it, but also files dynamically created by running programs. The "/tmp" directory is analogous to a WORK PACK in MVS.

If "/tmp" fills up, even commands like "NETSTAT" cease to work.

If "/tmp" is not used for extensive logging and tracing activity, you might be able to afford to leave it on a Temporary File System (TFS) that is cleaned up at every IPL.

A TFS is a file system stored in memory instead of dataset on DASD. A TFS loses all its data at every IPL or if it is unmounted.

A TFS delivers a high I/O rate because it is using a data space that is part of the kernel address space.

In any case, whatever mountpoint you use for logging, you should ensure that that directory never fills up. Otherwise you could lose important log data.

Even if you use an alternate mountpoint for logging (e.g., "/tmplog"), you must still periodically monitor /tmp to ensure that it doesn't fill up.

SYS1.PARMLIB(BPXPRMnn) to Define and Mount Temporary File System (TFS)

```
FILESYSTYPE TYPE(TFS) ENTRYPOINT(BPXTFS) /* TFS for /tmp */
MOUNT FILESYSTEM('/TMP') TYPE(TFS) /* temp space at /tmp */
MOUNTPOINT('/tmp') PARM('-s 80')
```

/etc/rc Definition to Create Target Log Files for SYSLOG Daemon and to Start

Create target files for SYSLOG daemon if it is running on TFS

```
#>/tmp/auth.log
```

```
#>/tmp/error.log
```

```
#>/tmp/debug.log
```

```
#>/tmp/syslog.log
```

Start the SYSLOG daemon for logging UNIX activity

```
_BPX_JOBNAME='SYSLOGD' /usr/sbin/syslogd -f /etc/syslog.conf &
```

```
# /usr/sbin/syslogd -f /etc/syslog.conf &
```

```
sleep 5
```

```
echo /etc/rc script executed, `date`
```

Display Unix File Usage

- df

```
- ***** Top of Data *****
- Mounted on      Filesystem      Avail/Total      Files      Status
- /u/users        (OMVS.HOMEDIRS.HFS)  12904/12960      4294967292 Available
- /u/tf           (OMVS.TF.HFS)        1400/1440        4294967294 Available
- /u/rdm          (OMVS.RDM.HFS)       177032/177120    4294967293 Available
- /u/jc           (OMVS.JC.HFS)        1400/83520       4294967051 Available
- /u/harris1      (OMVS.HARRISL.HFS)   176976/177120    4294967288 Available
- /u/gdente       (OMVS.GDENTE.HFS)    161552/177120    4294967238 Available
- /usr/lpp/HOD    (OMVS.HOD40.HOM.HFS) 222968/1308960   4294918223 Available
- /tmp            (OMVS.NM2.TMP)       174152/182880    4294967208 Available
- /etc            (OMVS.V2R7.ETC.HFS)  6792/11520       4294966979 Available
- /               (OMVS.V2R7.PUT9904.BASE.HFS) 131696/1398240  4294953323 Available
- ***** Bottom of Data *****
```

- df -P /tmp

```
- ***** Top of Data *****
- Filesystem      512-blocks      Used      Available      Capacity Mounted on
- OMVS.NM2.TMP    182880          8736      174144          5% /tmp
- ***** Bottom of Data *****
```

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 29

If you are managing "/tmp" manually, you may use the OMVS command "df" to view the space usage on the dataset mounted at /tmp.

"df" displays the amount of free space in the file system.

Default is to display in 512-byte units; "df -k" displays in 1024-byte (1KB) units.

To display usage on Hierarchical File Systems, use the OMVS command:

df (shows all usage)

df -P /tmp (shows usage for dataset mounted at /tmp mountpoint)

df -v (shows above plus file system type, mode bits, and mount parm data)

Format of display:

- File system root
- File System Name
- Space available and total space in already allocated extents
- Number of free files (inodes).

Number is accurate for files created with DFSMS 1.3.0 and later

For earlier versions of DFSMS, number is always 4,294,967,295

Maximum number is (2 to the 32nd power -1). Each file and (sub)directory is assigned an inode number. 4,294,967,292 means that there are three inodes associated with this filesystem.

Capacity refers to space utilized by the file system for disk user data and for metadata (attributed, inode numbers, space maps, etc) PLUS the space for metadata shadow writes to provide recovery capability for the dataset should the system fail during the update of the inodes.

The capacity number is not the same as that shown for ISPF. ISPF capacity does not include the extra space for shadow writes.

SMS Managed Volume

Data Set Information

Command ==>

Data Set Name . . . : OMVS.NM2.TMP

General Data

Management class . . . : STANDARD

Storage class . . . : SCOMVS

Volume serial . . . : NM27AF

Device type . . . : 3390

Data class . . . :

Organization . . . : PO

Record format . . . : U

Record length . . . : 0

Block size . . . : 0

1st extent cylinders: 5

Secondary cylinders : 1

Data set name type : HFS

Current Allocation

Allocated cylinders : 127

Allocated extents . . : 123

Maximum dir. blocks : NOLIMIT

Current Utilization

Used pages . . . : 1,176

% Utilized . . . : 5

Number of members . . : 40

Creation date . . . : 1998/05/29

Referenced date . . : 1999/10/21

Expiration date . . : ***None***

- Even though the usage percentage is not of concern, the limit of 123 extents has been reached.

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 30

Even though "/tmp" shows up as only 5% full with the "df -P" command, you could still have a space problem. Note how the dataset already has reached 123 extents, the maximum allowable number of extents for single-volume files. This dataset needs to be unloaded and reloaded with a larger primary extent.

DSSDUMP Job:

```
//Jobcard .....
//STEP1 EXEC PGM=ADDRSSU,REGION=6M
//SYSPRINT DD SYSOUT=*
//HFSOUT DD DSN=OMVS.NM2TMP.UNLOAD,DISP=(,CATLG),
//      SPACE=(CYL,(20,20),RLSE),VOL=SER=CNMCAT
//SYSIN DD *
DUMP DATASET(INCLUDE(OMVS.NM2.TMP)) -
  COMPRESS TOL(ENQF) -
  OUTDDNAME(HFSOUT)
/*
```

DSSREST Job:

```
//Jobcard .....
//STEP1 EXEC PGM=ADDRSSU,REGION=6M
//SYSPRINT DD SYSOUT=*
//HFSSEQ DD DSN=OMVS.NM2TMP.UNLOAD,DISP=SHR
//SYSIN DD *
RESTORE INDD(HFSSEQ) TOL(ENQF) -
DATASET(INCLUDE(OMVS.NM2.TMP)) -
CANCELERROR
/*
```

ISPF Space of 5% utilized is: space utilized by the file system for disk userdata and for metadata (attributed, inode numbers, space maps, etc)

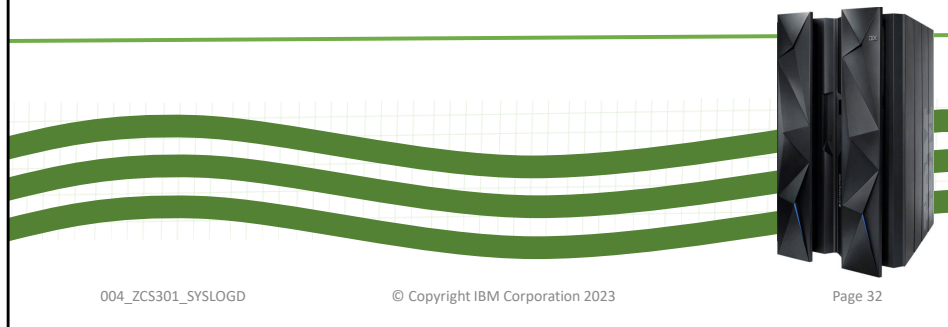
SMS Managed Volume

```
Esxxxxxxxxxxxxxxxxxxxxx VTOC Summary Information xxxxxxxxxxxxxxxxxxxxxxxxN
e Volume . : NM27AF e
e Command ==> e
e e
e Unit . . : 3390 e
e e
e Volume Data VTOC Data Free Space Tracks Cyls e
e Tracks . : 50,085 Tracks . : 29 Size . . : 717 47 e
e %Used . : 98 %Used . . : 5 Largest . : 285 19 e
e Trks/Cyls: 15 Free DSCBS: 1,379 e
e Free Extents . : 8 e
e e
e F1=Help F2=Split F3=Exit F9=Swap F12=Cancel e
DxxxxxxxxxxxxxxxxxxxxxM
```

- Total Used = 98%

If you run into storage problems, you must also check the total usage of the SMS-managed volume. Here you see that our storage problems are caused not only by having grown to the maximum number of allowable extents but also by having utilized nearly the entire volume (98% Used).

Time Stamps

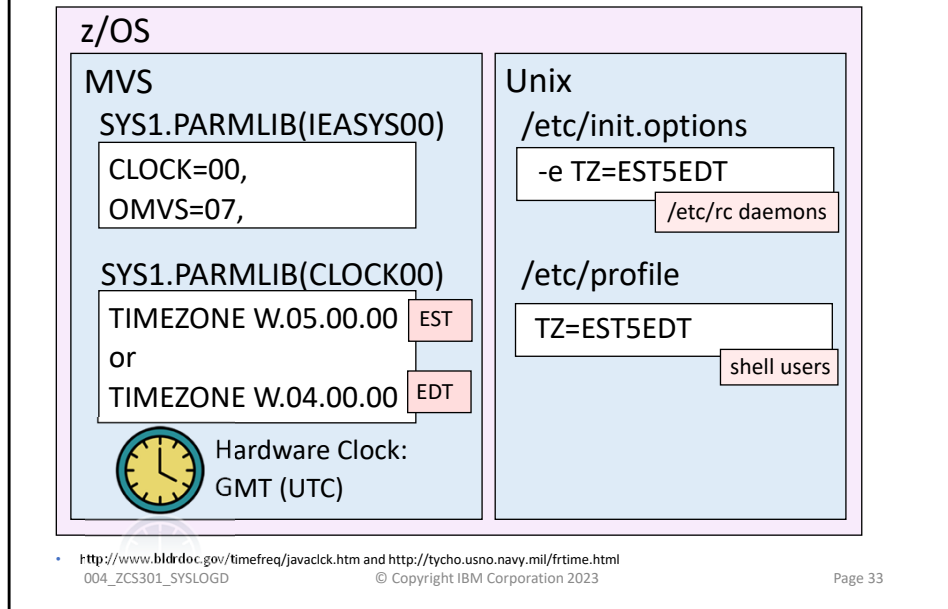


004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 32

Setting Time in z/OS



IBM recommends that your system hardware clock (processor timer) be set to UTC time. (UTC time is also known as GMT, or Greenwich Mean Time, because Greenwich, England is the site of the Royal Observatory, located at longitude zero.

From z/OS MVS Setting Up a Sysplex (SA22-7625), Chapter 2: Planning Parmlib Members for a Sysplex:

"To ensure that MVS makes time-related decisions as you intend, IBM recommends that you do not reset your TOD clock, for example, to switch to or from Daylight Savings Time. Instead, set the TOD clock to a standard time origin, such as Coordinated Universal Time (UTC), and use the TIMEZONE statement of the CLOCKxx parmlib member to adjust for local time. To adjust local time, change the offset value for the TIMEZONE statement. This will not disturb MVS's assumption that time is progressing forward and will allow time stamps on printed output and displays to match local time."

When you migrate to a UNIX-based z/OS system, you may observe that MVS applications display the local time but some UNIX System Services applications display the GMT time or even an incorrect time.

This may occur when you have not synchronized your UNIX timezone settings (TZ) with the OS/390 timezone settings (CLOCK). Note that the offset specified in the "CLOCK" member may apply either to the processor timer (ETRMODE=NO in the CLOCK member) or to the Sysplex Timer (ETRMODE=YES in the CLOCK member).

Or, it may occur because certain applications (like TCP/IP) do not consult the TZ settings in UNIX System Services.

Most daemons use the TZ setting in /etc/init.options which affects the initialization of processes invoked from /etc/rc at OMVS initialization; shell users use settings in /etc/profile (sets system-wide user environment) or the \$HOME/.profile (which can override the settings in /etc/profile).

SYSLOGD Log (Time not Synchronized)

```
Jul 8 15:24:03 WSC1 FSUM1220 syslogd: restart
Jul 8 19:25:53 WSC1 Config[67108868]: EZZ0300I OPENED PROFILE FILE
Jul 8 19:25:53 WSC1 Config[67108868]: EZZ0309I PROFILE PROCESSING
.....
Jul 8 19:28:11 WSC1 ftpd[369098755]: EZYFT18I Using catalog
Jul 8 19:28:11 WSC1 ftpd[369098755]: EZYFT08W Unable to get port
Jul 8 19:28:11 WSC1 ftpd[369098755]: EZY2697I IBM FTP CS V2R7
Jul 8 19:28:12 WSC1 ftpd[369098755]: EZY2640I Using
Jul 8 19:28:12 WSC1 ftpd[369098755]: EZYFT47I dd:SYSFTPD file,
.....
Jul 8 19:28:12 WSC1 ftpd[1577058316]: EZY2702I Server-FTP:
Jul 8 19:28:12 WSC1 ftpd[1577058316]: EZYFT41I Server-FTP: process
Jul 8 15:36:15 WSC1 inetd[83886093]: FOMN0044 Unable to lock /etc/inetd.pid:
EDC5112I Resource temporarily unavailable., rsn=055501B7
Jul 8 15:39:12 WSC1 inetd[134217741]: FOMN0026 otelnet/tcp: unknown service
Jul 8 15:47:25 WSC1 telnetd[33554448]: IP address is 9.82.131.114
```

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 34

Unless you manipulate the TZ (timezone) settings in your UNIX and MVS procedures, you can have mismatched Timestamps in the SYSLOG.

You see here that SYSLOGD itself is using the local time, whereas the FTP Server is using GMT time.

If you look at the syslog.log unix file for SysLogD, you'll notice that for things like TCP/IP's subcomponent Config, all the trace entries appear with GMT time stamps. This can be annoying and confusing when automating or sorting.

NETSTAT (Time not Synchronized)

====> netstat home

GMT or UDC Time

MVS TCP/IP NETSTAT CS V2R7 TCPIP NAME: NM2ATCP 21:49:42

Home address list:

Address	Link	Flg
---------	------	-----

-----	----	---
-------	------	-----

192.168.251.1	VLINK1	
---------------	--------	--

192.168.253.1	VLINK2	
---------------	--------	--

9.82.1.170	TR1	P
------------	-----	---

9.82.67.170	LNK2BTCP	
-------------	----------	--

Local (CLOCKnn=TIMEZONE W.04.00.00)

TIME-05:50:46 PM. CPU-00:00:05 SERVICE-663221 SESSION-01:49:17 JULY 14,2008

Synchronize Time Zones – Best Practice

- **SYS1.PARMLIB(CEEPRMxx)**
 - CEEDOPT(ALL31(ON), ENVAR('TZ=EST5EDT'))
 - CEECOPT(ALL31(ON), ENVAR('TZ=EST5EDT'))
 - CELQDOPT(ALL31(ON), ENVAR('TZ=EST5EDT'))
- Default Parmlib CEEPRMxx is found in CEE.SCEESAMP.

SYSLOGD Log (Time Synchronized)

```

Jul  8 08:31:34 LO0 FSUM1220 syslogd: restart
Jul  8 12:33:47 LO0 ConfigY16777218": EZZ0300I OPENED PROFILE FILE
Jul  8 12:33:48 LO0 ConfigY16777218": EZZ0316I PROFILE PROCESSING
Jul  8 12:33:48 LO0 ConfigY16777218": EZZ0334I IP FORWARDING IS
Jul  8 12:33:48 LO0 ConfigY16777218": EZZ0335I ICMP WILL IGNORE
Jul  8 12:33:48 LO0 ConfigY16777218": EZZ0352I VARIABLE SUBNETTING
Jul  8 12:33:48 LO0 ConfigY16777218": EZZ0345I STOPONCLAWERROR IS
Jul  8 12:34:03 LO0 ConfigY16777218": EZZ0403I TELNET/VTAM (SECOND
Jul  8 12:34:04 LO0 ftpdY13": EZYFT18I Using catalog '/usr/lib/nls
Jul  8 12:34:04 LO0 ftpdY13": EZY2697I IBM FTP CS V2R7 12:34:04
Jul  8 12:34:04 LO0 ftpdY13": EZY2640I Using dd:SYSFTPD=SYS1.TCPC
Jul  8 12:34:04 LO0 ftpdY13": GU0754 chkunit: unitname 3390
Jul  8 12:34:04 LO0 ftpdY13": EZYFT21I Using catalog '/usr/lib/nls
Jul  8 12:34:06 LO0 snmpagentY16": EZZ6202I Using catalog 'snmpd
Jul  8 12:34:06 LO0 snmpagentY16": EZZ6232I The SNMP agent is run
Jul  8 12:34:06 LO0 snmpagentY16": EZZ6295I SNMP agent: Dynamic
.....
Jul  8 12:37:49 LO0 telnetdY167772177": EYZTE52E Couldn't resolve
Jul  8 12:37:49 LO0 telnetdY167772177": IP address is 9.82.1.107

```

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 37

There are several ways to solve the Time Stamp problem:

1. Prior to z/OS V1R7 you could set TZ value in CEEBINIT and reassemble it:
 - ENVAR(('TZ=EST5EDT'),OVR)
2. Starting in z/OS V1R7 you could set TZ in CEEPRMxx:
 - CEEDOPT(ALL31(ON, ENVAR('TZ=EST5EDT')))
 - CEECOPT(ALL31(ON, ENVAR('TZ=EST5EDT')))
 - CELQDOPT(ALL31(ON, ENVAR('TZ=EST5EDT')))
3. Or defining TZ in all the daemon separately.


```

//FTPT21 PROC MODULE='FTPD',PARMS="
//FTPT21 EXEC PGM=&MODULE,REGION=4096K,TIME=NOLIMIT,
//  PARM=('POSIX(ON) ALL31(ON)',
//  'ENVAR("_BPXK_SETIBMOP_T_TRANSPORT=TCPT21",',
//  "'_BPX_JOBNAME=FTPT21'",
//  "'TZ=EST5EDT')/ &PARMS')
and
//OMPRT21 PROC
//OMPROUTE EXEC PGM=OMPROUTE,REGION=0K,TIME=NOLIMIT,
// PARM=('POSIX(ON)',
//  'ENVAR("_CEE_ENVFILE=DD:STDENV",',
//  "'_CEE_RUNOPTS=HEAP(,,,FREE)'",
//  "'TZ=EST5EDT')/ ')

```

Time zone could be defined in the Standard Environment file for the application instead of in the JCL procedure.

NETSTAT (Time Synchronized)

====> netstat home

MVS TCP/IP NETSTAT CS V2R7 TCPIP NAME: NM2ATCP 18:04:34

Home address list:

Address	Link	Flg
---------	------	-----

-----	----	---
-------	------	-----

192.168.251.1	VLINK1	
---------------	--------	--

192.168.253.1	VLINK2	
---------------	--------	--

9.82.1.170	TR1	P
------------	-----	---

9.82.67.170	LNK2BTCP	
-------------	----------	--

TIME 06:05:30 PM. CPU-00:00:05 SERVICE-773160 SESSION-02:04:01 JULY 14,1999

Appendix: Cron Daemon

CRON was popular prior to Automatic Archive option.
Use Automatic Archive or CRON, not both!



004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 39

z/OS Cron definitions are documented in the following manuals:

z/OS UNIX System Services Planning

z/OS UNIX System Services User's Guide

z/OS UNIX System Services Command Reference

CRON

- CRON Daemon is a work scheduler for Unix in z/OS.



004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 40

Cron is the Greek word for Time.
Chronos is the Greek Titan God of Time.

Files used by CRON

- /etc/mailx.rc file
 - Only required for cron to send messages via sendmail.
- /usr/lib/cron directory contains:
 - at.allow – contains the list of users who have permission to use at command
 - at.deny - contains the list of users who do not have permission to use at command
 - cron.allow - contains the list of users who have permission to use crontab command
 - cron.deny - contains the list of users who do not have permission to use crontab command
- /usr/spool/cron directory contains:
 - log - file that maintains a history of the commands being run
 - pid - file that cron uses to ensure that only one version of cron is currently running
 - queuedefs – defines queue values
- /usr/spool/cron/atjobs directory contains:
 - at files
- /usr/spool/cron/crontabs directory contains:
 - crontab files (cron job files)

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 41

queuedefs file contains lines with syntax:

[q].[njob]j[nice]n[nwait]w

where the fields are:

- q Specifies the name of the queue.
 - a - default queue
 - b - batch queue
 - c - crontab files

Queue names can be any single-byte character except a space, tab, newline, null, or number sign (#).

- njob Specifies the maximum number of jobs that can be run in the queue simultaneously. The default value is 100.
- nice Specifies the nice value (like UID). The default value is 2.
- nwait Specifies the number of seconds that cron is to wait before it reschedules a job. The default value is 60.
- Lines beginning with a number sign (#) are comments, and are ignored.

##

Sample queuedefs file

#

a.5j3n

b.3j1n90w

Where

a.5j3n q=a,njob=5,nice=3,nwait defaults to 60

b.3j1n90w q=b,njob=3,nice=1,nwait=90

Start CRON Daemon

- /etc/rc

```
# Start the SYSLOG daemon for logging UNIX activity
_BPX_JOBNAME='SYSLOGD' /usr/sbin/syslogd -f /etc/syslog.conf &
# Start the INET daemon for remote login activity
_BPX_JOBNAME='INETD' /usr/sbin/inetd /etc/inetd.conf &
# Start the CRON daemon for automated, timed operations
_BPX_JOBNAME='CRON' /usr/sbin/cron &
#
sleep 5
echo /etc/rc script executed, `date`
```

- D OMVS,A=ALL
OMVSKERN CRON7 0033 16777221 1 1KI 08.24.48 .053
LATCHWAITPID= 0 CMD=/usr/sbin/cron

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 42

Although you can start the CRON Daemon with JCL, it will come up automatically at OMVS startup if you place the startup information in the /etc/rc file as you see above.

You can code the startup in two ways:

- Use _BPX_JOBNAME to give the Daemon an MVS jobname, which you can see in the MVS system log. (See bottom of page.)
- Code the startup command for CRON without _BPX_JOBNAME. The job then appears in the MVS system log with a name associated with ETCRCn.

At the installation time, you have to define at least one superuser and group to which the superuser will belong. To control the different levels of superuser privileges, you can use several BPX RACF facility classes. The concept of superuser is expanded to include:

- A user with an effective UID of 0.
- A started task with trusted or privileged attribute defined in the RACF STARTED Class.

You may define the following three RACF facility classes to help you distinguish between different levels of superuser privileges:

- BPX.SUPERUSER
- DAEMON
- SERVER

If a server program needs to change User IDs in his process, the user ID that is assigned to this process must be permitted to READ the BPX.DAEMON facility class. As soon as the BPX.DAEMON facility class has been defined in RACF, a UID of zero is not sufficient for changing the user identity of a process. The process must run with a UID of zero and must be permitted to BPX.DAEMON to do so. Typically server processes (TelnetD, REXECD, FTPD) need to be able to change the Unix identity to the identity of the client user after having obtained a valid user ID and password. There are a number of configuration tasks that you must perform if you choose to use this level of security in your environment. Once you define the BPX.DAEMON facility class, you have to enable the program control for certain IP load libraries.

Authorize Userid with BPX.DAEMON:

- RDEFINE FACILITY BPX.DAEMON UACC(NONE)
- SETROPTS CLASSACT(FACILITY) GENERIC(FACILITY AUDIT(FACILITY))
- SETROPTS RACLIST(FACILITY)
- PERMIT BPX.DAEMON CLASS(FACILITY) ID(OMVSKERN) ACCESS(READ)

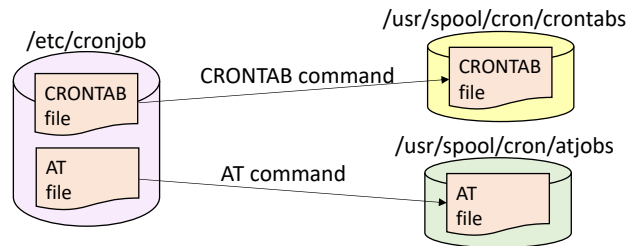
Facility Class BPX.DAEMON Restricts who can change the security context of a UNIX process with a valid password.

The FTP server and the UNIX versions of the TelnetD, REXECD, and RSHD servers require READ access. CRON Daemon also requires BPX.DAEMON authority.

Example definitions:

- RDEFINE FACILITY BPX.DAEMON UACC(NONE)
- SETROPTS CLASSACT(FACILITY) GENERIC(FACILITY) AUDIT(FACILITY)
- SETROPTS RACLIST(FACILITY)
- PERMIT BPX.DAEMON CLASS(FACILITY) ID(OMVSKERN) ACCESS(READ)

CRON Commands



- Three different commands for submitting CRON Files:
 - CRONTAB – for CRON files defining repeating CRON jobs
 - AT – for CRON files defining one time CRON jobs
 - BATCH – for CRON files defining one time batch CRON jobs
- Implementation Steps
 - Create CRONTAB, AT, or BATCH file
 - Use CRONTAB, AT, or BATCH command to submit the file to CRON

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 43

`/etc/cronjob` is any path. Create CRONTAB, AT, or BATCH files in some user directory (ie. `/etc/cronjob`) and then use the CRONTAB, AT, or BATCH command to submit them to CRON. In this way you do not edit the files that CRON use directly, preventing accidental typos.

CRONTAB Job File

CRONTAB file syntax:

Mininue Hour Day Month DayOfWeek command

Minute Day Day of Week
↓ ↓ ↓
53 23 * * 0 echo Now replacing log files with the A set
↑ ↑ ↑
Hour Month Execution string

```
# This is a sample crontab file stored
# in my maintenance directory
52 23 * * 0 echo Now replacing log files with the A set
53 23 * * 0 cp /etc/syslog.conf.a /etc/syslog.conf
54 23 * * 0 kill -SIGHUP $(cat /etc/syslog.pid)
55 23 * * 0 cp /tmp/syslog.log.b /tmpback/syslog.log.backb
58 23 * * 0 rm /tmp/syslog.log.b
59 23 * * 0 touch /tmp/syslog.log.b
```

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 44

Day of week: 0

= Sunday; 1=Monday; etc.

CRON must be running for the CRONTAB command to work.

If you do not build a CRON.ALLOW or CRON.DENY, then no user may issue the CRONTAB command.

You may build an EMPTY CRON.DENY, which will allow any user to execute the CRONTAB command in certain flavors.

You may build a CRON.DENY file with certain users to be excluded from crontab authorization. Everyone else may enter "crontab..."

Alternatively, you may build a CRON.ALLOW file, which allows only the listed users to execute the command.

If you build the CRON.ALLOW file, ensure that you include the superuser OMVSKERN in the file as well as any operations staff, like GDENTE2. When GDENTE2 issues the shell command "crontab -l", he is shown any crontab entry that has his USERID name on it.

He is not a Superuser, so he may view only the crontab that has been filed under his userid.

USERIDs MUST BE IN UPPER CASE.

CHARLIE is not included in the cron.allow file and as a result does not pass validation when he enters the "crontab -l" command.

CRONTAB Command

- **CRONTAB [-u user] [filename] -- highly recommended!**
 - Copies file into CRON directory /usr/spool/cron/crontabs for use.
 - Superuser may define a specific user to associate the file with.
 - Recommended instead of CRONTAB -e which can inadvertently delete the crontab entry.
- **CRONTAB -l [-u user] -- to list your current crontab tasks**
- **CRONTAB -r [-u user] -- to delete a file from crontab subdirectory**
- **CRONTAB -e [-u user] – lets you edit your crontab entry.**
Define editor environment variable or crontab defaults to vi.
Not recommended - don't directly edit the file! You could inadvertently delete your entries.

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 45

The owner of the crontab file must execute the crontab command to store the file. The CRON Daemon verifies what the UID is of the userid under which the crontab file is stored and cross-checks it with the owner attribute of the stored file the daemon read from the CRONTABS subdirectory. If they do not match, you get an error message like:

- Security error: file "GDENTE2" owner is #0, should be #9
- This error occurred because a Superuser issued the crontab command to store the data from the user's backup directory into the crontabs subdirectory. GDENTE2 has a UID of 9, but the file named "GDENTE2" had an owning UID of 0.

Once the user named GDENTE2 issued the crontab command "crontab /u/gdente2/GDENTE2," the CRON daemon could handle the execution request without an error:

```
> CMD: echo Commit senseless acts of beauty!  
> OMVSKERN 352321544 c Wed Jul 14 19:38:00 1999  
< OMVSKERN 352321544 c Wed Jul 14 19:38:01 1999 rc=0
```

Any Superuser can store the Files with the USERIDs of other superusers, because they all have UID=0.

AT Command

- `at [-m] [-f file] [-q queue] -t time`
 - `at [-m] [-f file] [-q queue] timespec`
 - `at -r [-q queue] at_job ...`
 - `at -l [-q queue] [at_job ...]`
 - `-f file` Reads commands from *file* rather than from standard input (stdin).
 - `-l` Displays all jobs you have submitted with at command.
 - `at_job` Filters display to only show jobs with matching job name.
 - `-m` Sends you mail after your job has finished running.
 - `-q queue` Specifies the queue your at job is to be recorded in or removed from.
 - `-r at_job` Removes previously scheduled at jobs.
 - `-t time` Specifies the time for the system to run the job.
 - When you do not use the `-t` option, you can use a *timespec* argument to specify the time.
 - A *timespec* argument consists of three parts: a time, a date, and an increment. You must always specify the time, but you can omit the date, the increment, or both.
 - For time specification see the z/OS UNIX System Services Command Reference, SA23-2280
 - The **batch** command is equivalent to:
 - `at -q b -m now`
- 004_ZCS301_SYSLOGD © Copyright IBM Corporation 2023 Page 46

`at` lets you set up a series of commands to be run later. It reads the commands from the standard input (stdin) or from a file specified with the `-f` option.

`batch` lets you run commands in batch mode. It reads the commands from the standard input (stdin). The system records the commands and runs them at a time when the system load is relatively low (that is, when the system is not busy).

Using Cron Steps

- Start CRON in /etc/rc during OMVS startup:
Start Cron
_BPX_JOBNAME='CRON' /usr/sbin/cron &
- Create flat cron job file - such as /etc/cronjob which contains:
0 0 * * * kill -HUP `cat /etc/syslog.pid`
0 0 * * * /u/user1/rmoldlogs
- Execute crontab command to load cron job file:
crontab /etc/cronjob
- List cron jobs:
USER1:/u/user1: >crontab -l
0 0 * * * kill -HUP `cat /etc/syslog.pid`
0 0 * * * /u/user1/rmoldlogs

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 47

A user (usually a superuser) submits a work request.

If the user is authorized to use the appropriate shell command (at.allow, at.deny, cron.allow, cron.deny), the work request is stored in the appropriate file under the name of the user. (For CRONTAB: /usr/spool/cron/crontabs) If a superuser submits the request, the file is known as OMVSKERN.

Either at CRON startup or through a signal from the user (the crontab command is a signal), the cron daemon reads the stored work requests.

The cron daemon then builds or rebuilds the work queues. ALL requests found in the "/at" or "/crontab" subdirectory are processed.

The scheduler function of the cron daemon monitors the queues; when a designated time for a work request arrives, the scheduler makes a call to the system.

This causes Work Load Manager to fork an address space for the work request. The request is run under the UID of the requester. If only superusers are allowed to control cron through the crontab and "at" commands, you do not need to define the additional cron control files in the /usr/lib/cron directory:

- at.allow
- at.deny
- cron.allow
- cron.deny

Cron also uses the /usr/spool/cron directory for runtime files.

The five options before the command are:

- Minute of the hour
- Hour of the day
- Day of the month
- Month of the year
- Day of the week (Sunday=0)

CRON Log

- This is the cron log stored in /usr/spool/cron/log.

```
> CMD: echo Replace log files with A set and Copy B to MVS
> OMVSKERN 184549386 c Tue Jul 13 23:51:03 1999
< OMVSKERN 184549386 c Tue Jul 13 23:51:04 1999 rc=0
> CMD: /etc/replaceb.sh
> OMVSKERN 201326602 c Tue Jul 13 23:52:00 1999
< OMVSKERN 201326602 c Tue Jul 13 23:52:08 1999 rc=0
> CMD: echo If only the weekend could begin now!
> OMVSKERN 939524108 c Wed Jul 14 12:15:01 1999
< OMVSKERN 939524108 c Wed Jul 14 12:15:02 1999 rc=0
> CMD: echo Commit random acts of kindness!
> OMVSKERN 956301324 c Wed Jul 14 12:56:01 1999
< OMVSKERN 956301324 c Wed Jul 14 12:56:01 1999 rc=0
```

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 48

Day of week: 0

= Sunday; 1=Monday; etc.

CRON must be running for the CRONTAB command to work.

If you do not build a CRON.ALLOW or CRON.DENY, then no user may issue the CRONTAB command.

You may build an EMPTY CRON.DENY, which will allow any user to execute the CRONTAB command in certain flavors.

You may build a CRON.DENY file with certain users to be excluded from crontab authorization. Everyone else may enter "crontab..."

Alternatively, you may build a CRON.ALLOW file, which allows only the listed users to execute the command.

If you build the CRON.ALLOW file, ensure that you include the superuser OMVSKERN in the file as well as any operations staff, like GDENTE2. When GDENTE2 issues the shell command "crontab -l", he is shown any crontab entry that has his USERID name on it.

He is not a Superuser, so he may view only the crontab that has been filed under his userid.

USERIDs MUST BE IN UPPER CASE.

CHARLIE is not included in the cron.allow file and as a result does not pass validation when he enters the "crontab -l" command.

Mail Log

- Activity also shows up in /usr/mail/(username), as identified by the CRONTAB entry names.

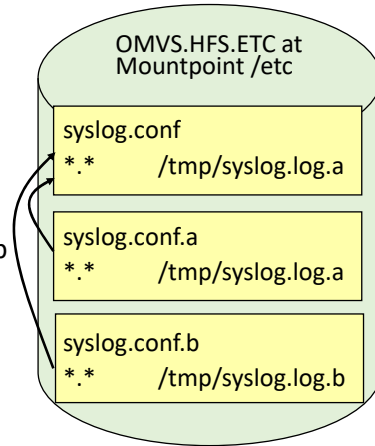
```
From OMVSKERN Tue Jul 13 23:52:10 1999
To: OMVSKERN

alloc da('gdente.syslog.log.b') dsorg(ps) space(3,1) cylinders 1
oget '/tmp/syslog.log.b' 'gdente.syslog.log.b'
BPXF112W THE RECORD SIZE IN THE OUTPUT DATA SET IS SMALLER THAN

*****
Cron: The previous message is the standard output
      and standard error of one of your cron commands.
```

Manage SYSLOGD with CRON

- Create “a” and “b” version of syslog configuration file.
- Day 1
 - cp syslog.conf.a syslog.conf
 - Messages are logged to /tmp/syslog.log.a
 - Optionally backup /tmp/syslog.log.b
 - Remove /tmp/syslog.b
- Day 2
 - cp syslog.conf.b syslog.conf
 - Messages are logged to /tmp/syslog.log.b
 - Optionally backup /tmp/syslog.log.a
 - Remove /tmp/syslog.log.a
- Repeat swapping files each day.



004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 50

SyslogD configuration file sample in /usr/lpp/tcpip/samples contains sample definitions for CRON usage as detailed here. You can't manipulate the active log file because syslogd is actively logging to it. So start logging to a new active log file first. Then you can backup and empty the original log file.

Sample CRONTAB File (unix to unix)

```
# Copies logs into HFS dataset
# ORIGINAL Crontab named OMVSKERN.hfs stored in
# /usr/spool/cron/crontabs
# Every night archive log except Saturday night
# Another process archives log files on /tmpback to tape or other
52 23 * * 0 echo Now replacing log files with the A set
53 23 * * 0 cp /etc/syslog.conf.a /etc/syslog.conf
54 23 * * 0 kill -SIGHUP $(cat /etc/syslog.pid)
55 23 * * 0 cp /tmp/syslog.log.b /tmpback/syslog.log.backb
58 23 * * 0 rm /tmp/syslog.log.b
59 23 * * 0 touch /tmp/syslog.log.b
#
52 23 * * 1 echo Now replacing log files with the B set
53 23 * * 1 cp /etc/syslog.conf.b /etc/syslog.conf
54 23 * * 1 kill -SIGHUP $(cat /etc/syslog.pid)
55 23 * * 1 cp /tmp/syslog.log.a /tmpback/syslog.log.backa
58 23 * * 1 rm /tmp/syslog.log.a
59 23 * * 1 touch /tmp/syslog.log.a
#
    AND SO ON THROUGH DAY 5!
```

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 51

```
#
53 23 * * 2 echo Now replacing log files with the A set
54 23 * * 2 cp /etc/syslog.conf.a /etc/syslog.conf
55 23 * * 2 kill -SIGHUP $(cat /etc/syslog.pid)
56 23 * * 2 cp /tmp/syslog.log.b /tmpback/syslog.log.backb
59 23 * * 2 rm /tmp/syslog.log.b
00 00 * * 2 touch /tmp/syslog.log.b
#
53 23 * * 3 echo Now replacing log files with the B set
54 23 * * 3 cp /etc/syslog.conf.b /etc/syslog.conf
.....
#
53 23 * * 4 echo Now replacing log files with the A set
54 23 * * 4 cp /etc/syslog.conf.a /etc/syslog.conf
.....
#
53 23 * * 5 echo Now replacing log files with the B set
54 23 * * 5 cp /etc/syslog.conf.b /etc/syslog.conf
.....
```

Sample CRONTAB File (unix to MVS)

```
# Copies logs into MVS dataset
# Every night archive log except Saturday night
# Another process archives log files to tape or other from MVS
#
52 23 * * 0 echo Now replacing log files with the A set
53 23 * * 0 cp /etc/syslog.conf.a /etc/syslog.conf
54 23 * * 0 kill -SIGHUP $(cat /etc/syslog.pid)
55 23 * * 0 tso -t "OGET '/tmp/syslog.log.b' 'GDENTE.SUNLOG.MVS'"
58 23 * * 0 rm /tmp/syslog.log.b
59 23 * * 0 touch /tmp/syslog.log.b
#
52 23 * * 1 echo Now replacing log files with the B set
53 23 * * 1 cp /etc/syslog.conf.b /etc/syslog.conf
54 23 * * 1 kill -SIGHUP $(cat /etc/syslog.pid)
55 23 * * 1 tso -t "OGET '/tmp/syslog.log.b' 'GDENTE.MONLOG.MVS'"
58 23 * * 1 rm /tmp/syslog.log.a
59 23 * * 1 touch /tmp/syslog.log.a
#
# AND SO ON THROUGH DAY 5!
```

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 52

```
#
53 23 * * 2 echo Now replacing log files with the A set
54 23 * * 2 cp /etc/syslog.conf.a /etc/syslog.conf
55 23 * * 2 kill -SIGHUP $(cat /etc/syslog.pid)
.....
#
53 23 * * 3 echo Now replacing log files with the B set
54 23 * * 3 cp /etc/syslog.conf.b /etc/syslog.conf
55 23 * * 3 kill -SIGHUP $(cat /etc/syslog.pid)
.....
#
53 23 * * 4 echo Now replacing log files with the A set
54 23 * * 4 cp /etc/syslog.conf.a /etc/syslog.conf
55 23 * * 4 kill -SIGHUP $(cat /etc/syslog.pid)
.....
#
53 23 * * 5 echo Now replacing log files with the B set
54 23 * * 5 cp /etc/syslog.conf.b /etc/syslog.conf
55 23 * * 5 kill -SIGHUP $(cat /etc/syslog.pid)
.....
```

Appendix: Cron Example



004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 53

Sample CRONTAB File (unix to MVS)

```
# Copies logs into MVS dataset
# Every night archive log except Saturday night
# This CRONTAB = /etc/OMVSKERN.scr2
# Another process archives the log files from MVS to GDS and/or tape
#
51 23 * * 0 echo Replace log files with A set & Copy B to MVS
52 23 * * 0 /etc/replaceb.sh
#
51 23 * * 1 echo Replace log files with B set & Copy A to MVS
52 23 * * 1 /etc/replacea.sh
#
51 23 * * 2 echo Replace log files with A set & Copy B to MVS
52 23 * * 2 /etc/replaceb.sh
#
51 23 * * 3 echo Replace log files with B set & Copy A to MVS
52 23 * * 3 /etc/replacea.sh
#
#          AND SO ON THROUGH DAY 5!
```

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 54

CRONTAB Shell Script (unix to MVS)

```
# Step 1) Setting up Variables
# LOG_DIRECTORY is the path of the syslogs files. i.e. /tmp/syslog
LOG_DIRECTORY='/tmp'
# The following variables hold the names of the various log files
SYSLOG_LOGA='syslog.log.a'
# ERROR_LOG='error.log'
# DS_PREFIX is the Data Set Prefix. Files will be named with this hlq
# For example, HFS error.log becomes 'gdente.error.log' in MVS
DS_PREFIX='gdente'
# The following loop iterates through all the log files and executes
# the commands in the loop on each file.
#for LOGFILE in $SYSLOG_A $ERROR_LOG
for LOGFILE in $SYSLOG_LOGA
do
#
# Step 2) Allocating MVS Datasets
#
tso -t "alloc da('$DS_PREFIX.$LOGFILE')dsorg(ps)space(3,1) cylinders \
lrecl(132) blksize(13200) recfm(f,b) volume(csscat) unit(sysda) old"
#lrecl(132) blksize(13200) recfm(f,b) volume(csscat) unit(sysda) new"
#
```

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 55

```
# Shell Script To Copy A unix Log Files to MVS Datasets /etc/replacea.sh
# This script swaps out the syslogd logs in unix file and
# then writes them to an MVS dataset, from which an automated batch
# job can transfer them into a Generation Dataset or archive them
# otherwise. This shell runs on alternating days with another
# shell script named /etc/replaceb.sh. The MVS batch automation
# job is not provided here.
# STEPS:
# 1) Set up variables.
# 2) Allocate a file for the MVS dataset; the file is allocated
# as "old" and so it must exist prior to this step. You
# may run this shell script first with a "new" allocation
# as indicated below. All subsequent allocations must be
# with the "old" designation. See examples below.
# 3) Copy B configuration file into /etc/syslog.conf
# to record on B logs
# 4) Force SYSLOGD to reread the new configuration file
# 5) Copy the old A logs to an MVS dataset and
# wait one minute
# 6) Delete and recreate the empty A logfile
# NOTE: Another archiving method should archive the MVS
# datasets within 2 days before they are replaced with
# the next reallocation ("old") of the same dataset.
```

CRONTAB Shell Script (unix to MVS)

```
# Step 3) Swap out syslogd.conf files (Copy B configuration file
#         into /etc/syslog.conf to record on B logs)
#
cp /etc/syslog.conf.b /etc/syslog.conf
#
# Step 4) Force SYSLOGD to reread the new configuration file
#
kill -SIGHUP $(cat /etc/syslog.pid)
# Step 5) Copy old A logs to an MVS Dataset and wait 1 minute
sleep 1
tso -t oget '$LOG_DIRECTORY/$LOGFILE\' \'$DS_PREFIX.$LOGFILE\'
# Step 6) Delete and Recreate the A log file
#         We've copied the specified file if it exists,
#         so now we should delete and recreate the log file.
rm $LOG_DIRECTORY/$LOGFILE
touch $LOG_DIRECTORY/$LOGFILE
#
# DONE - files should now be in a MVS dataset for some other archiver
# to handle. Every night, with the exception of Saturday, the cron
# daemon uses the crontab entry to swap out the log files. Archiver
# program must run at least every two days; otherwise the data is
# overwritten with a new tso allocate command ("old").
done
```

004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 56

IMPORTANT:

```
# The script is submitted from within a crontab entry that is
# maintained in the /etc directory. The crontab entry is
# /etc/OMVSKERN.scr2. It is stored in the /usr/spool/cron/crontabs
# directory by a superuser invoking the shell command
# "crontab /etc/OMVSKERN.scr2"
# Once the superuser has stored the crontab entry, he can view it via
# "crontab -l"
# To view cron log activity, browse
# /usr/spool/cron/log
# WARNING: Do not directly edit the crontab entry within the
# /usr/spool/cron/crontabs directory. Results are unpredictable.
# Always edit from the /etc/ directory and store the results with
# "crontab /etc/OMVSKERN.scr2"
#
```

End of Topic



004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 57

End of Topic



004_ZCS301_SYSLOGD

© Copyright IBM Corporation 2023

Page 58