

CheatSheet #59 zTidBits z/OS Logs and the Logger

Do not overlook log data — it should be the first place to look when reviewing a problem.

- z/OS communicates problems through messages that it writes to logs.

- Six logs contain the primary sources of problem data and information on system activity:

SYSLOG: The SYSLOG is a SYSOUT data set provided by the job entry subsystem (either JES2 or JES3).

>SYSLOG data sets are output spool data sets on direct access storage devices (DASD).

>An installation should print the SYSLOG periodically to check for problems.

The SYSLOG consists of:

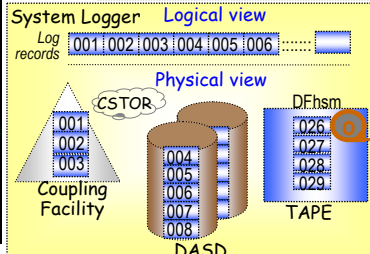
1. All messages issued through WTL macros (SVC 36 Write-To-Log)
2. All messages entered by LOG operator commands
3. Typically, the hard-copy log
4. Any messages routed to the SYSLOG from any system component or program

See: z/OS Diagnosis Reference GA22-7588 & Auth. Assem. Services GA22-7612

View SYSLOG through the **Spool Display and Search Facility (SDSF)** using the LOG option (See illustration).

- A small amount of the SYSLOG is also stored in memory and is included when an address space is dumped.
- This is the **master trace (MTRACE)** data and can be accessed from IPCS* using the VERBX MTRACE command.

The master trace table entries for the dumped system. This table is a wraparound data area that holds the most recently issued console messages in a first-in, first-out order.



Job log Messages sent to the job log are intended for the programmer who submitted a job.

- Specify the system output class for the job log in the MSGCLASS parameter of the JCL JOB statement.

NOTE: MSGCLASS=x is used to specify where the job output will be directed. The output classes are predefined and vary from site to site.

OPERLOG Operations log (OPERLOG) is an MVS system logger application that records and merges messages about programs and system functions (the hardcopy message set) from each system in a sysplex that activates OPERLOG.

- In SDFS the OPERLOG panel displays the merged, sysplex-wide system message log.

- You can use the parameters of the LOG command to select the OPERLOG panel or the single-system SYSLOG panel.

- The OPERLOG panel displays the data from a log stream, a collection of log data used by the MVS System Logger to provide the merged, sysplex-wide log.

- An individual product has its own log file.

- These log files might contain data that is valuable when diagnosing a problem.

- It is particularly important to look for events that precede an actual abend or failure because the problem, in many cases, will have been caused by a previous action.

-The key SYSOUT data sets to review for problem determination data are the JESMSGLG and MSGUSR data sets.

-The CEEMSG and CEEOUT data sets will contain Language Environment® (LE) problem data typically associated with application problems.

-The CICS JESMSGLG SYSOUT data set includes information related to CICS startup and errors related to system problems, not specifically transaction related.

Logrec Error Recording Log recording (logrec) log stream is an MVS System Logger application that records hardware errors, selected software errors, and symptom records across the sysplex.

- Use the records in the logrec data set or the logrec log stream as additional information when a dump is produced.
- The information in the records can point you in the right direction while supplying you with symptom data about the failure.
- Use the Environmental Record, Editing, and Printing program (EREP) to:

Applications can use the IXGBRWSE service to read and browse a log stream for log block information.

*IPCS - Interactive Problem Control System (dump analysis.)

- Logrec data is written to the SYS1.LOGREC data set and is also written to internal storage that is included in a dump.

- The SYS1.LOGREC data set can be interrogated using the ICFEREP1 program, or if the abend has triggered a dump, the EREP data can be reviewed using the IPCS VERBX LOGDATA command.

Generally, the error log entries at the end of the display, if they have an influence on the problem being reviewed, have time stamps that relate to or immediately precede the actual abend; although there is no guarantee the error records will be written in the order they occurred.

- The error log entries are also written to an internal storage buffer that is included in the dump.

- Using a logrec log stream rather than a logrec data set (SYS1.LOGREC, by default) for each system can streamline logrec error recording.

Console log Console data that the installation specifically chooses to log.

Hardcopy log The hardcopy log is a record of the system message traffic that the installation chooses to log, such as messages to and from 'all' consoles, commands and replies entered by the operator (used for auditing).

- In a dump, these messages are in the master trace and with JES3, the hardcopy log is always written to the SYSLOG.

- With JES2, the hardcopy log is typically written to the SYSLOG, but can also be written to a console printer, if your installation chooses.

System Logger is an z/OS component that provides a logging facility for applications running in a single-system or multi-system sysplex.

- The advantage of using System Logger is that the responsibility for tasks such as saving the log data (with the requested persistence), retrieving the data (potentially from any system in the sysplex), archiving the data, and expiring the data is removed from the creator (user, application or subsystem) of the log records.

- Logger provides the ability to have a single, merged, log containing log data from multiple instances of an application within the sysplex.

- Log data managed by the System Logger may reside in processor storage, in a Coupling Facility structure, on DASD, or potentially on tape.

- However, regardless of where System Logger is currently storing a given log record, from the point of view of the exploiter, all the log records are kept in a single file that is a limited size.

- The location of the data, and the migration of that data from one level to another, is transparent to the application and is managed completely by System Logger, with the objective of providing optimal performance while maintaining the integrity of the data (see logical/physical illustration on left).

- The task of tracking where a specific piece of log data is at any given time is handled by System Logger.

- System Logger will manage the utilization of its storage - as the space in one medium starts filling up (a Coupling Facility structure, for example), Logger will move old data to the next level in the hierarchy (dfhsm).

- By providing these capabilities using a standard interface, many applications can obtain the benefits that System Logger provides without having to develop and maintain these features themselves.

- This results in faster development, more functionality, and better reliability.

- Enhancements to System Logger, such as support for System Managed CF Structure Duplexing, become available to all System Logger exploiters as soon as they are implemented in System Logger, rather than having to wait for each exploiter to design, write, and test their own support.

How System Logger is used: There are basically two types of users of System Logger.

- Some exploiters basically use System Logger as an archival facility for log data.

> These exploiters dump their log data into System Logger and rely on it to manage the archival and expiration of the data from their own.

> These exploiters have the ability to subsequently retrieve the data should they need to do so, but their normal mode of operation would be to just give data to System Logger and not look for it back again.

Example: CICS Forward Recovery logs, where CICS stores data away in case a forward recovery is required some time in the future. **NOTE:** We call these exploiters **funnel-type exploiters**.

- The other type of exploiter typically uses the data more actively, and explicitly deletes it when it is no longer required.

Example: The CICS DFHLOG where CICS stores information in DFHLOG about running transactions, and deletes the records as the transactions complete. **NOTE:** We call these types of exploiters **active exploiters**.

As you can imagine, the performance requirements of these exploiters will differ. The exploiters that use Logger primarily to archive data are not particularly concerned about retrieval speeds, whereas an active user of the data will ideally want all the active data to be kept in a high performance location, like a data space or a CF structure.

Where System Logger stores its data: When an application passes log data to System Logger, the data can initially be stored on DASD, in what is known as a **DASD-only log stream**, or it can be stored in a Coupling Facility (CF) in what is known as a **CF-Structure log stream**.

- The major differences between these two types of log stream configurations are the storage medium System Logger uses to hold interim log data, and how many systems can use the log stream concurrently.

- In a CF log stream, interim storage for log data is in CF list structures supporting the ability for exploiters on more than one system to write log data to the same log stream concurrently.

- In a DASD-only log stream, interim storage for log data is contained in a **data space** in the z/OS system.

> The data spaces are associated with the System Logger address space, IXGLOGR. **See #54 zTidBits Extend. Address.**

> DASD-only log streams can only be used by exploiters on one system at a time.

System Logger: A component of the operating system that provides logging services. This component is commonly referred to as the MVS System Logger, z/OS System Logger, or simply as System Logger or just Logger. While there are enhancements that have been introduced by specific releases of the operating system, these terms are generally used interchangeably, irrespective of the release or version of the operating system being used.

Log streams: A sequence of data blocks, with each log stream identified by its own **log stream identifier**—the log stream name (LSN). Log streams data can span multiple recording media: interim (primary) storage, secondary (DASD based) storage, and tertiary (that is, tape migrated) media.

Interim Storage: Interim storage is the primary storage used to hold log data that has not yet been off loaded. The interim storage medium used depends on how the log stream has been defined; it may be a Coupling Facility (CF) structure or a staging data set. Log data that is in interim storage is duplexed to prevent against data loss conditions.

LOGR Couple Data Set: The LOGR Couple Data Set (CDS) holds the **System Logger policy** information and information about all defined log streams, and must be accessible by all the systems in the sysplex.

Log stream definitions: The log streams are defined using the **IXCMAPU program**, and the definitions are stored in the System Logger policy in the LOGR CDS.

Terms and Definitions