MVS logger: push your logs up the stream

Monday, February 24 2:30pm

Annapolis 2

Mike Shorkend

mike@shorkend.com

IBM Champion 2025

IBM Champion

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

IBM*
IBM Z *
IBM logo*
IMS
DB2
CICS
z/OS*

* Registered trademarks of IBM Corporation

* All other products may be trademarks or registered trademarks of their respective companies.

About me

- *Independent Consultant*
- *35+ years of Mainframe Infrastructure Experience*
- *Experience includes z/OS system programming,capacity planning and control, performance tuning, software pricing, architecture, BCP/DRP, software development and support, management training, mentoring.*
- *Customers include financial institutes, government agencies, VARs, ISVs and training companies*
- *SHARE volunteer since 2018*

Agenda

- Introduction
- Defining logstreams
- Major keywords for defining logstreams
- z/OS logger exploiters
- Miscellanea

INTRODUCTION

What is the MVS Logger?
What are logstreams?

- Logstreams were first introduced with Sysplex in 1995.
- A data structure that provides a mechanism of high performance/high volume logging
- The MVS logger is the z/OS facility that takes care of this
- An address space – IXGLOGR performs the actions against logstreams
- Does not require a Sysplex, can be implemented in a single image

What can you do with a logstream?

- Use system macros to access logstreams. These are the common ones:
  - **IXGCONN — Connect/disconnect to log stream**
  - **IXGWRITE — Write log data to a log stream**
  - **IXGBRWSE — Browse/read a log stream**
  - **IXGDELET — Deleting log data from a log stream**
- All macros are here
- A sample program(that reads OPERLOG) is here:

SAMPLIB(IEAMDBLG)

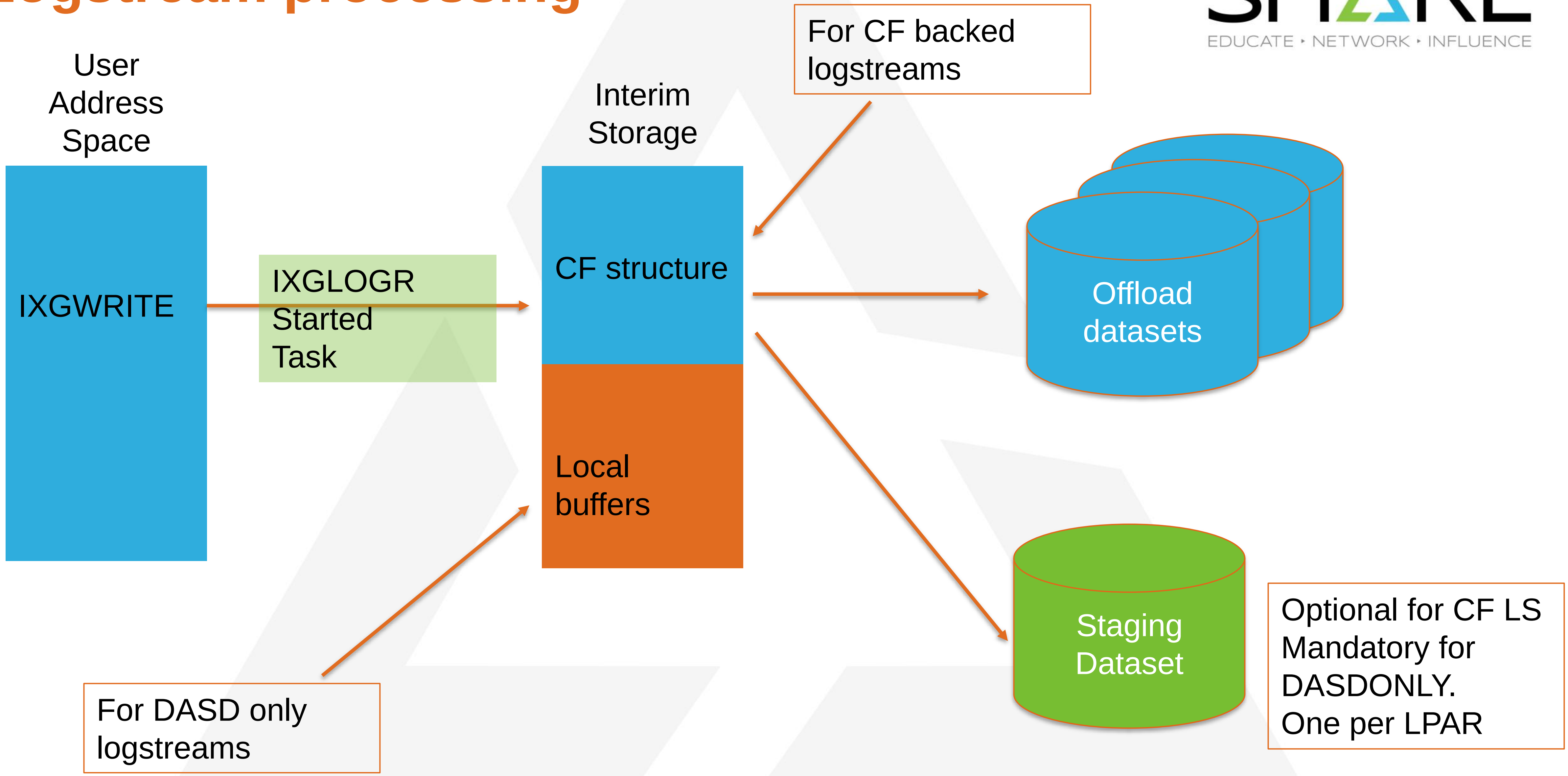
Who uses logstreams

- z/OS components: OPERLOG, LOGREC,SMF,RRS,CEA,Health Checker
- CICS log journals for recovery
- IMS shared queues
- VSAM logging
- Third party products that require high performance logging
- Homegrown applications

Two kinds of logstreams

- Coupling facility structure(not DASDONLY)
  - Requires a CF structure as interim storage
  - Provides logstream sharing between LPARs(e.g. OPERLOG)
- DASDONLY
  - Only one LPAR can use it
  - Requires a staging dataset

Logstream processing



User Address Space

IXGWRITE

IXGLOGR Started Task

Interim Storage

CF structure

Local buffers

For CF backed logstreams

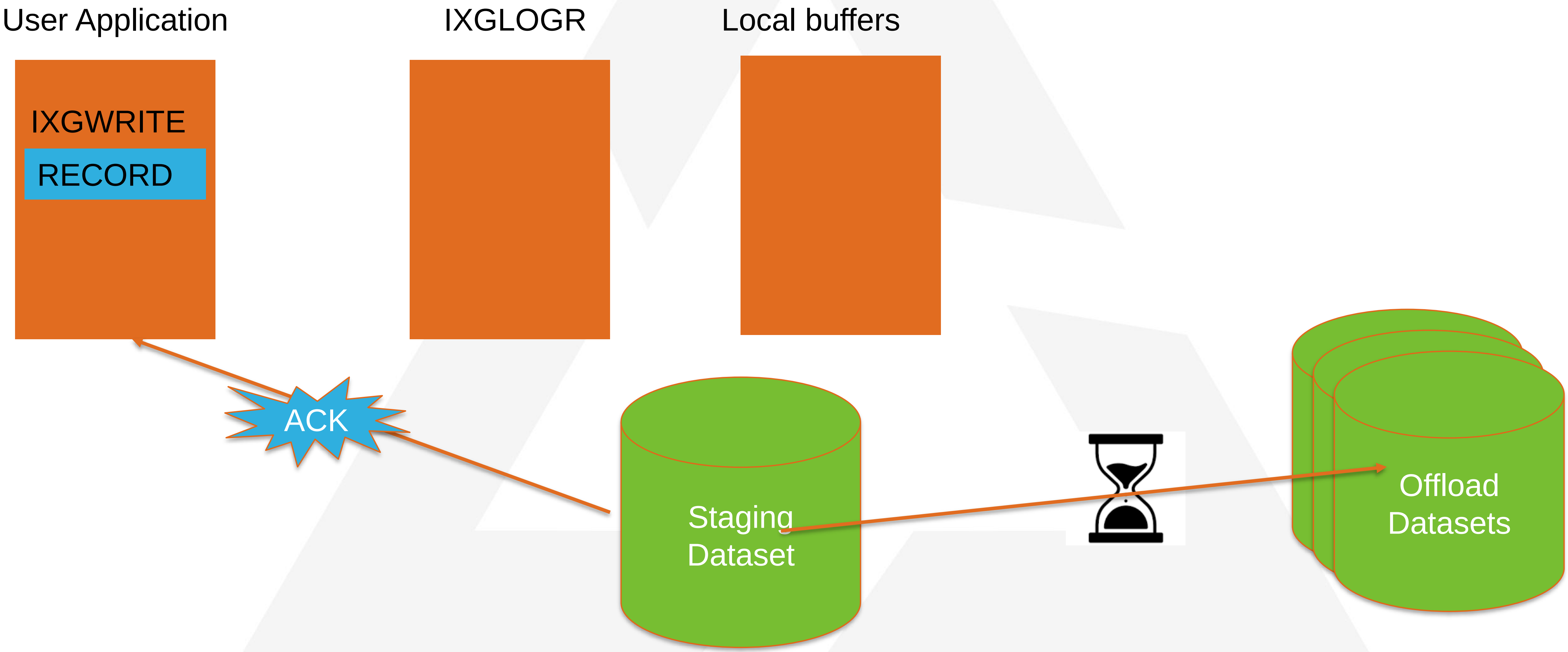
Offload datasets

Staging Dataset

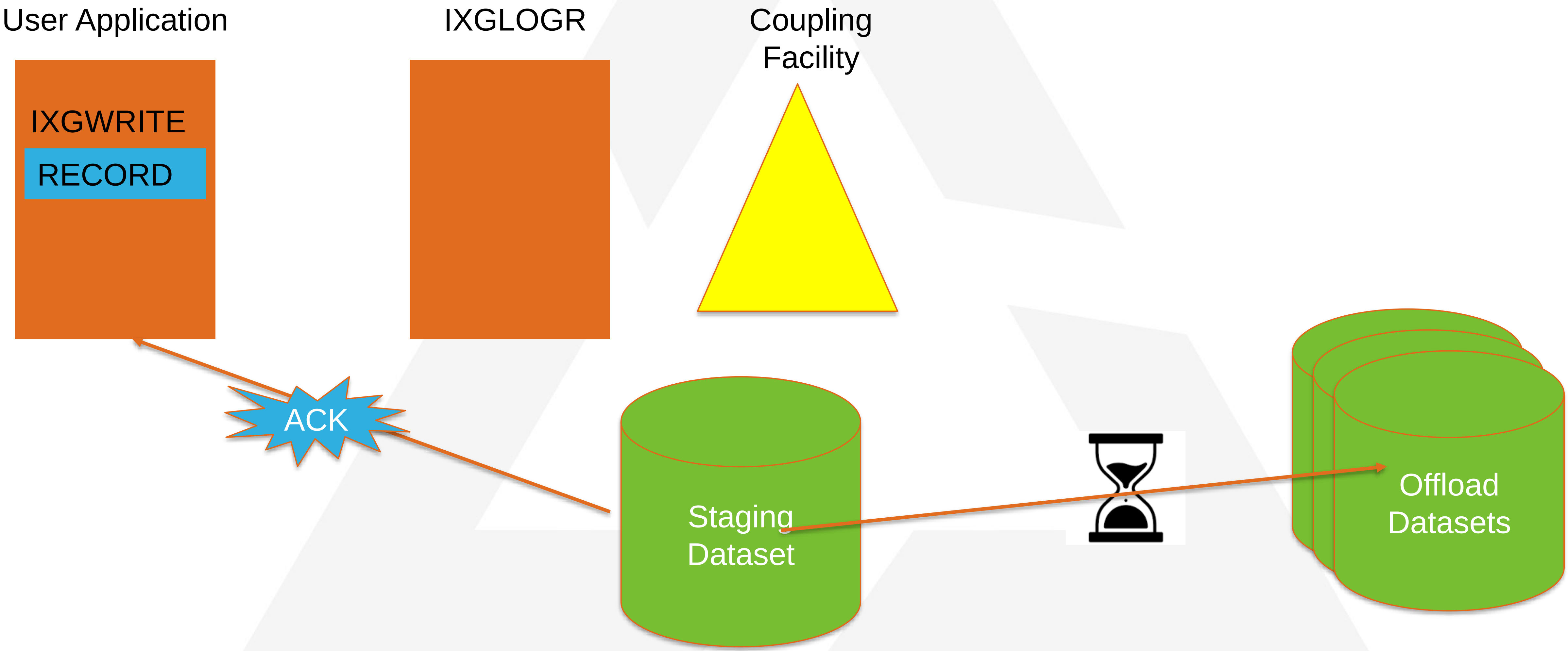
Optional for CF LS
Mandatory for DASDONLY.
One per LPAR

For DASD only logstreams

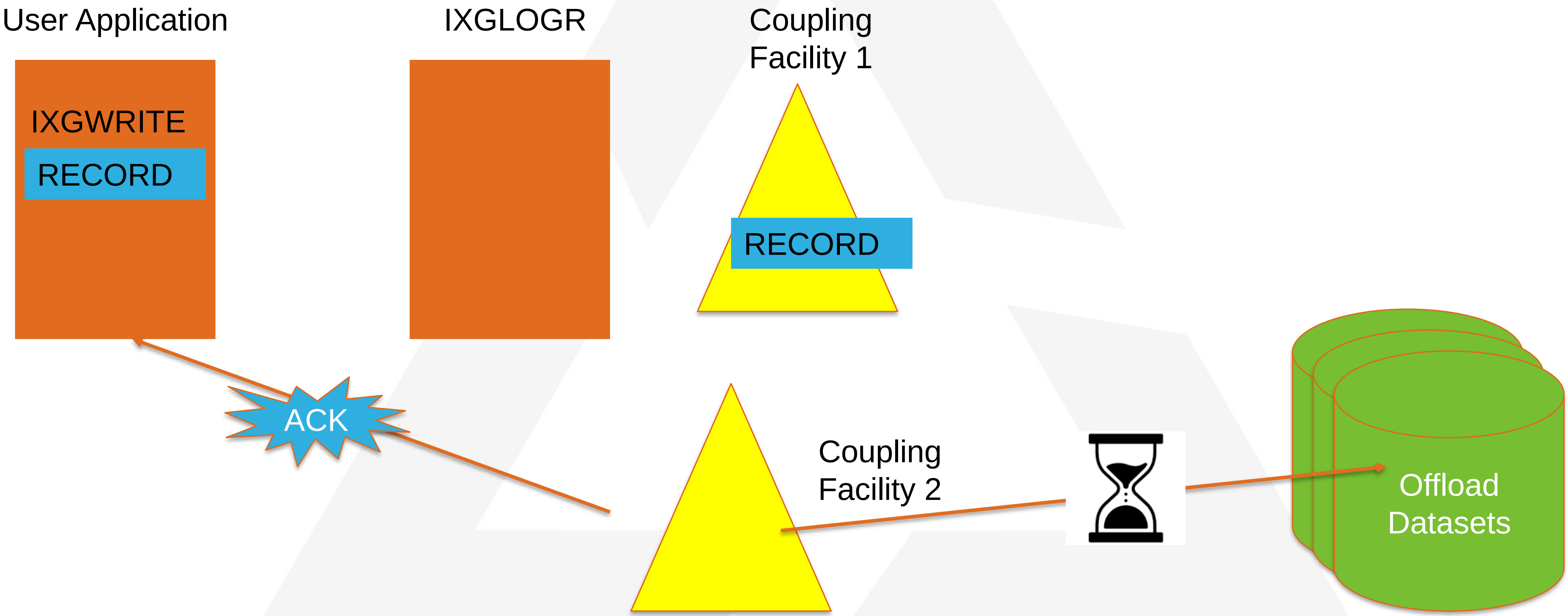
Duplexing – DASD ONLY



Duplexing – CF backed Logstream Option 1
Logger duplexes



Duplexing – CF backed Logstream Option 2 System duplexess



DEFINING LOGSTREAMS

Defining logstreams

- Logstreams need a LOGR couple dataset
- Use the **IXCL1DSU** utility to format using

```
          DATA TYPE(LOGR)
```

- Define them to COUPLEXX
- Each logstream needs to be defined to the LOGR policy using the **IXCMIAPU** utility
- CF structure backed logstreams need a structure

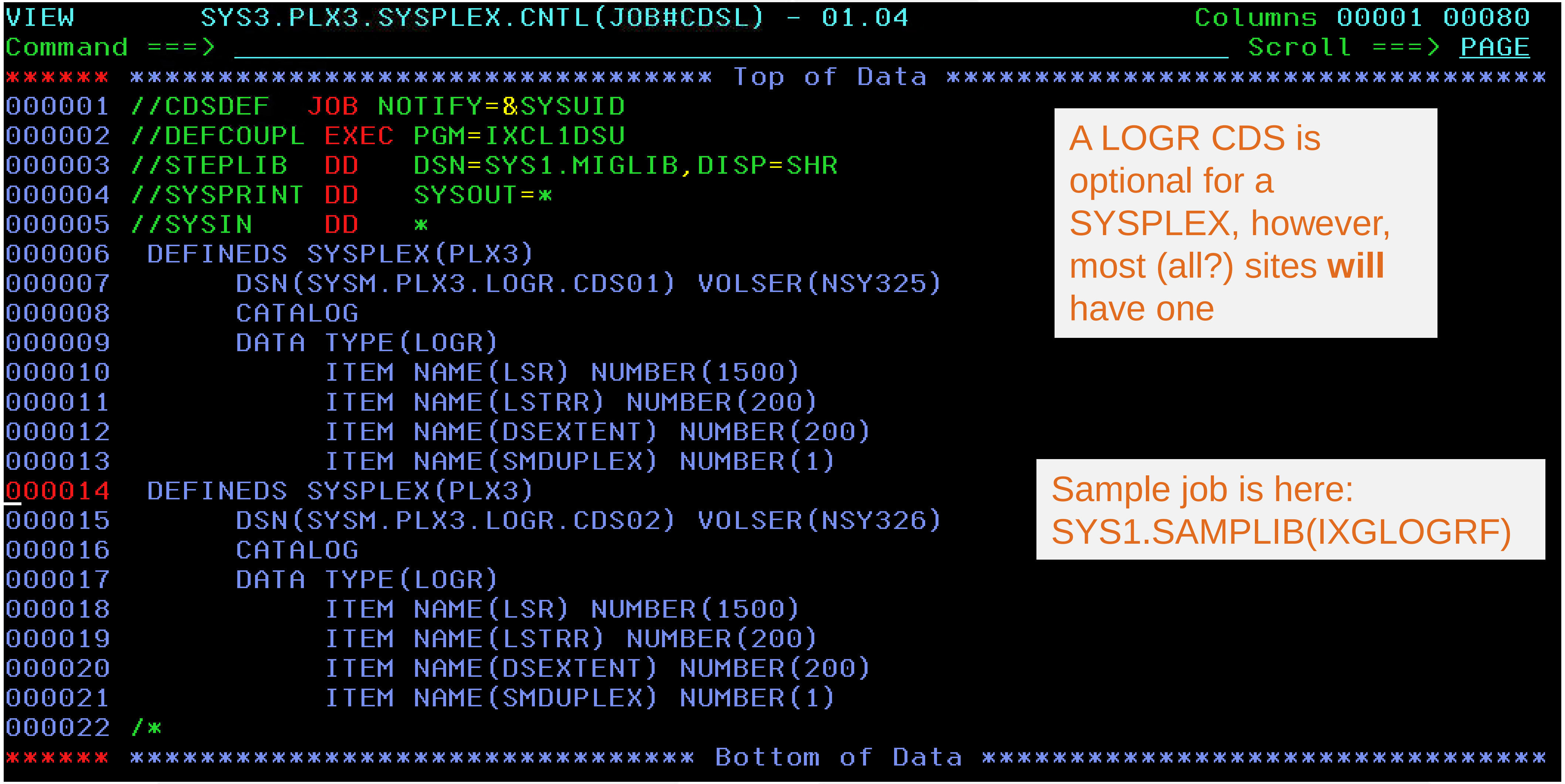
```
     DEFINE STRUCTURE NAME(STRUCTURE1) . . .
```

- Both kinds need a logstream definition

```
     DEFINE LOGSTREAM NAME(STREAM1). . .
```

- Setting up a Sysplex

Defining the LOGR couple dataset

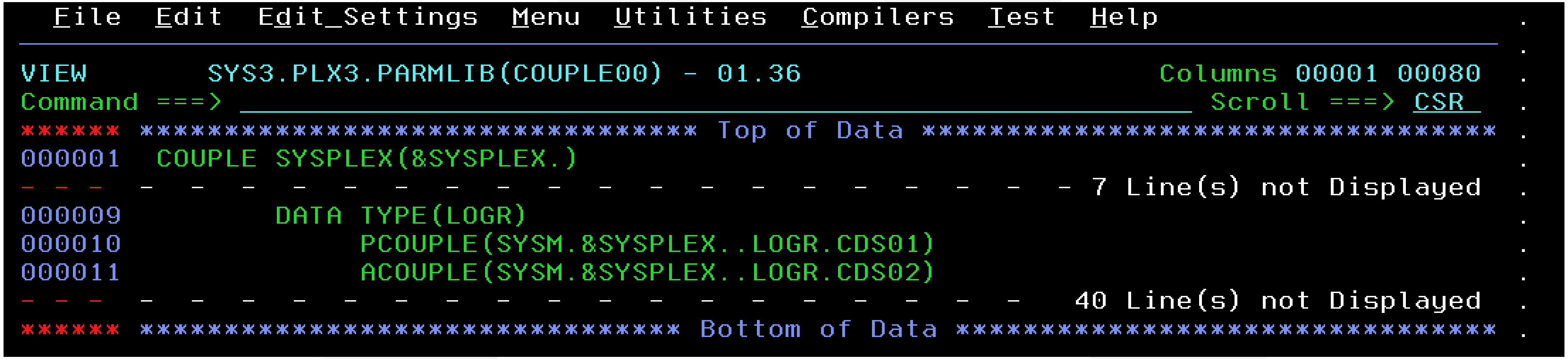


```
VIEW       SYS3.PLX3.SYSPLEX.CNTL(JOB#CDSL) - 01.04            Columns 00001 00080
Command ===> ________________________________________________________ Scroll ===> PAGE
****** ********************************* Top of Data **********************************
//CDSDEF   JOB NOTIFY=&SYSUID
//DEFCOUPL EXEC PGM=IXCL1DSU
//STEPLIB  DD   DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 DEFINEDS SYSPLEX(PLX3)
      DSN(SYSM.PLX3.LOGR.CDS01) VOLSER(NSY325)
      CATALOG
      DATA TYPE(LOGR)
           ITEM NAME(LSR) NUMBER(1500)
           ITEM NAME(LSTRR) NUMBER(200)
           ITEM NAME(DSEXTENT) NUMBER(200)
           ITEM NAME(SMDUPLEX) NUMBER(1)
 DEFINEDS SYSPLEX(PLX3)
      DSN(SYSM.PLX3.LOGR.CDS02) VOLSER(NSY326)
      CATALOG
      DATA TYPE(LOGR)
           ITEM NAME(LSR) NUMBER(1500)
           ITEM NAME(LSTRR) NUMBER(200)
           ITEM NAME(DSEXTENT) NUMBER(200)
           ITEM NAME(SMDUPLEX) NUMBER(1)
/*
****** ******************************** Bottom of Data ********************************
```

A LOGR CDS is optional for a SYSPLEX, however, most (all?) sites **will** have one

Sample job is here: SYS1.SAMPLIB(IXGLOGRF)

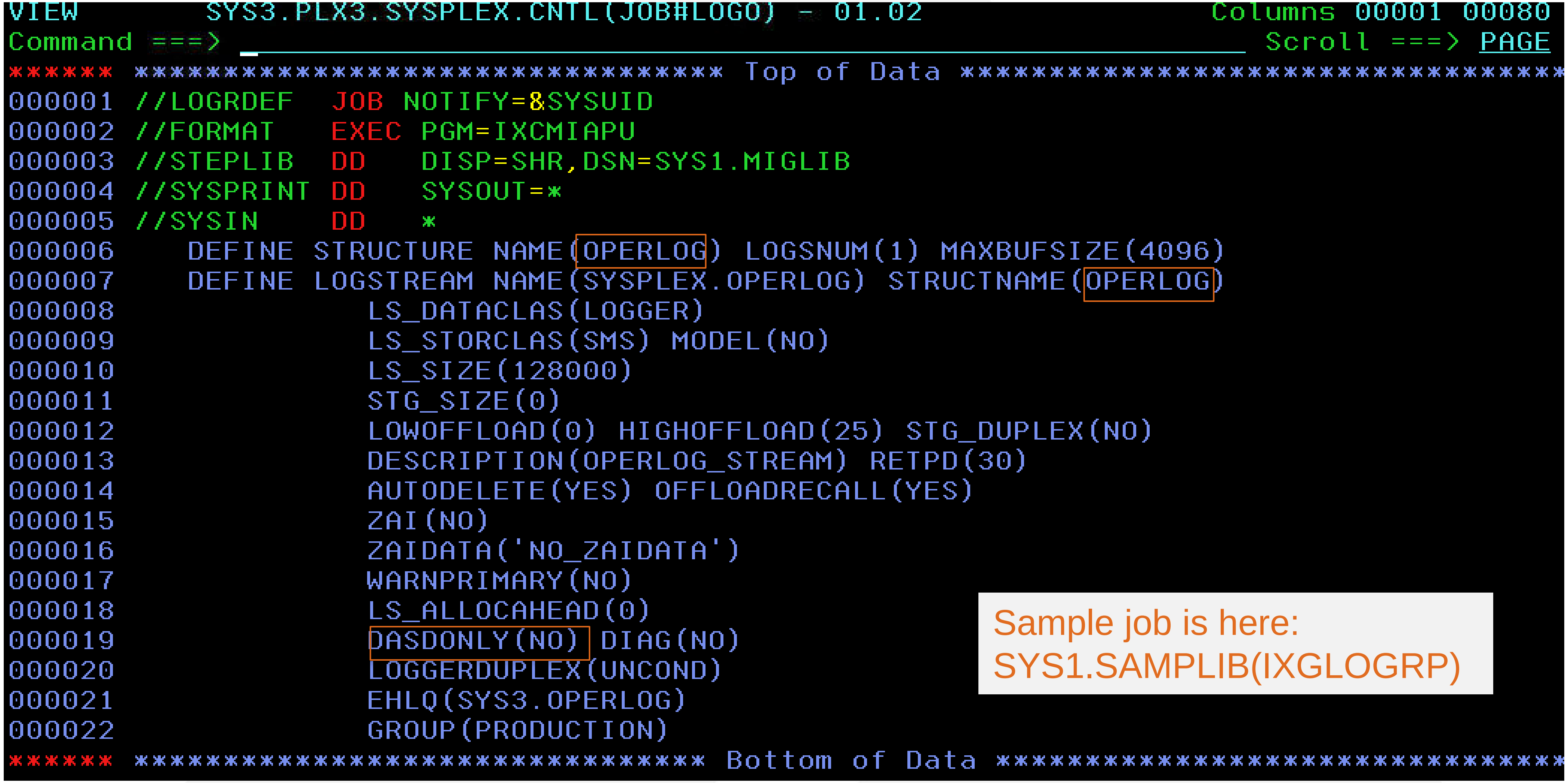
Defining the LOGR couple data set to the system



```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help

VIEW       SYS3.PLX3.PARMLIB(COUPLE00) - 01.36                    Columns 00001 00080
Command ===>                                                         Scroll ===> CSR
****** ******************************** Top of Data **********************************
000001  COUPLE SYSPLEX(&SYSPLEX.)
 - - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 7 Line(s) not Displayed
000009         DATA TYPE(LOGR)
000010              PCOUPLE(SYSM.&SYSPLEX..LOGR.CDS01)
000011              ACOUPLE(SYSM.&SYSPLEX..LOGR.CDS02)
 - - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  40 Line(s) not Displayed
****** ******************************** Bottom of Data ********************************
```

Defining a structure backed logstream



```
VIEW       SYS3.PLX3.SYSPLEX.CNTL(JOB#LOGO) - 01.02             Columns 00001 00080
Command ===> ________________________________________________________ Scroll ===> PAGE
****** ******************************** Top of Data ******************************
       //LOGRDEF   JOB NOTIFY=&SYSUID
       //FORMAT    EXEC PGM=IXCMIAPU
       //STEPLIB   DD   DISP=SHR,DSN=SYS1.MIGLIB
       //SYSPRINT DD    SYSOUT=*
       //SYSIN     DD   *
          DEFINE STRUCTURE NAME(OPERLOG) LOGSNUM(1) MAXBUFSIZE(4096)
          DEFINE LOGSTREAM NAME(SYSPLEX.OPERLOG) STRUCTNAME(OPERLOG)
                    LS_DATACLAS(LOGGER)
                    LS_STORCLAS(SMS) MODEL(NO)
                    LS_SIZE(128000)
                    STG_SIZE(0)
                    LOWOFFLOAD(0) HIGHOFFLOAD(25) STG_DUPLEX(NO)
                    DESCRIPTION(OPERLOG_STREAM) RETPD(30)
                    AUTODELETE(YES) OFFLOADRECALL(YES)
                    ZAI(NO)
                    ZAIDATA('NO_ZAIDATA')
                    WARNPRIMARY(NO)
                    LS_ALLOCAHEAD(0)
                    DASDONLY(NO) DIAG(NO)
                    LOGGERDUPLEX(UNCOND)
                    EHLQ(SYS3.OPERLOG)
                    GROUP(PRODUCTION)
****** ******************************** Bottom of Data ****************************
```

Sample job is here:
SYS1.SAMPLIB(IXGLOGRP)

PARAMETERS

Some important format parameters (IXCL1DSU)

- LSR – total number of logstreams that can be defined to the Sysplex.
- LSTRR – number of LOGR structures that can be defined to the Sysplex
- DSXTENT – provides additional space for offload datasets that need more than 168 datasets
- SMDUPLEX – allow for system duplexing of LOGR structures

Some important define logstream parameters (IXCMIAPU)1/2

- DASDONLY – structure backed or not
- STRUCTNAME – name of the structure
- MAXBUFSIZE – for DASDONLY (on structure def for CF backed) – the maximum blocksize written to DASD
- STG_DUPLEX/DUPLEXMODE/LOGGERDUPLEX – duplexing parameters, more on this later
- STG_DATACLAS/STORCLAS/MGMTCLAS- overrides for staging datasets
- LS_DATACLAS/STORCLAS/MGMTCLAS- overrides for offload datasets.
- LS_SIZE,STG_SIZE – the size in 4K blocks of the offload and staging datasets, respectively

Some important define logstream parameters 2/2

- LS_ALLOCAHEAD – the number of offload datasets (1 to 3) to be pre-allocated before they are needed.
- AUTODELETE and RETPD – if and when to delete logstream records automatically(more on deletion later)
- HLQ/EHLQ – the high level qualifier for the offload dataset(more later)
- HIGHOFFLOAD/LOWOFFLOAD – when to start and stop offloading records from the structure to the offload dataset, based on structure occupancy

Duplexing – DASD only logstreams

Parameter	Meaning	Value	
STG_DUPLEX	Copy logger data to staging Datasets	NO	Forced to YES
		YES	Duplexed to staging datasets
DUPLEXMODE	When to Duplex	COND	Forced to UNCOND
		UNCOND	
LOGGERDUPLEX	Should Logger or CF duplex?	UNCOND	Logger will DUPLEX
		COND	Forced to UNCOND

Some more information here

Duplexing – CF backed logstreams

Parameter	Meaning	Value	
STG_DUPLEX	Copy logger data to staging Datasets	NO	Duplexed in local buffers only. *Can lead to data loss*
		YES	Duplexed to staging datasets
DUPLEXMODE	When to Duplex	COND	Only if there is a SPOF
		UNCOND	Always
LOGGERDUPLEX	Should Logger or CF duplex?	UNCOND	Logger will DUPLEX
		COND	Use CF Duplexing

The importance of some parameters

- One of my customers is an ISV who provides a CDC solution for IMS
- It picks ups changes that are committed to IMS and writes them to a logstream.
- I wanted to test the importance (from a performance perspective) of the sizing parameters.
- My test - ran a batch job which drove 200,000 updates to the logstream
  - With a well defined logstream, LS_SIZE(25600) , STG_SIZE(256000) – each write took 2 msecs
  - With a badly defined logstream LS_SIZE(16) , STG_SIZE(16) – 12 msecs!

Testing log data set parameter modifications

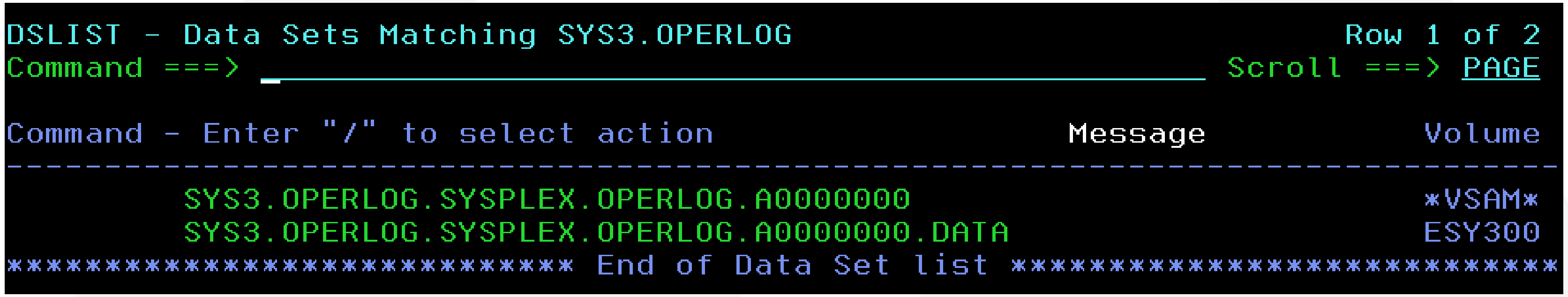
Logstream dataset naming 1/2

- Logstream datasets are VSAM linear datasets
- The high level qualifier for logstream datasets is determined by the HLQ or EHLQ parameters. The HLQ is up to eight characters in length, the EHLQ is up to 33. Normal dataset name standards. HLQ and EHLQ are mutually exclusive.
- The offload datasets look like this:

(E)HLQ.logstreamname.A0000nnnn

- Example:

```
DEFINE LOGSTREAM NAME(SYSPLEX.OPERLOG) EHLQ(SYS3.OPERLOG)...
```



```
DSLIST - Data Sets Matching SYS3.OPERLOG                            Row 1 of 2
Command ===> _                                                   Scroll ===> PAGE

Command - Enter "/" to select action                  Message           Volume
-------------------------------------------------------------------------------
         SYS3.OPERLOG.SYSPLEX.OPERLOG.A0000000                          *VSAM*
         SYS3.OPERLOG.SYSPLEX.OPERLOG.A0000000.DATA                     ESY300
***************************** End of Data Set list ****************************
```

Logstream dataset naming 2/2

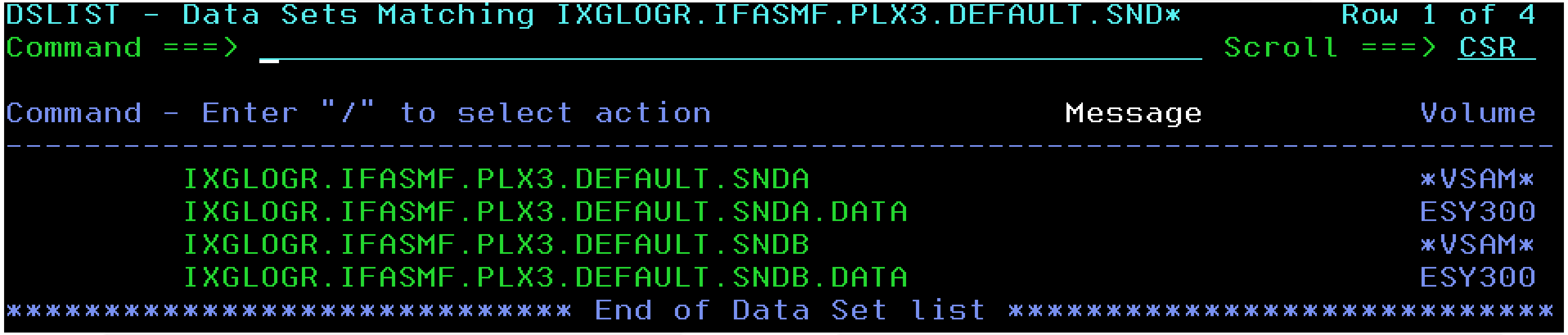
- There is a staging dataset for each LPAR in the Sysplex. Staging datasets look like this:

(E)HLQ.logstreamname.lparname

Example:

```
DEFINE LOGSTREAM NAME(IFASMF.PLX3.DEFAULT)
```

(E/HLQ omitted so it defaults to IXGLOGR)

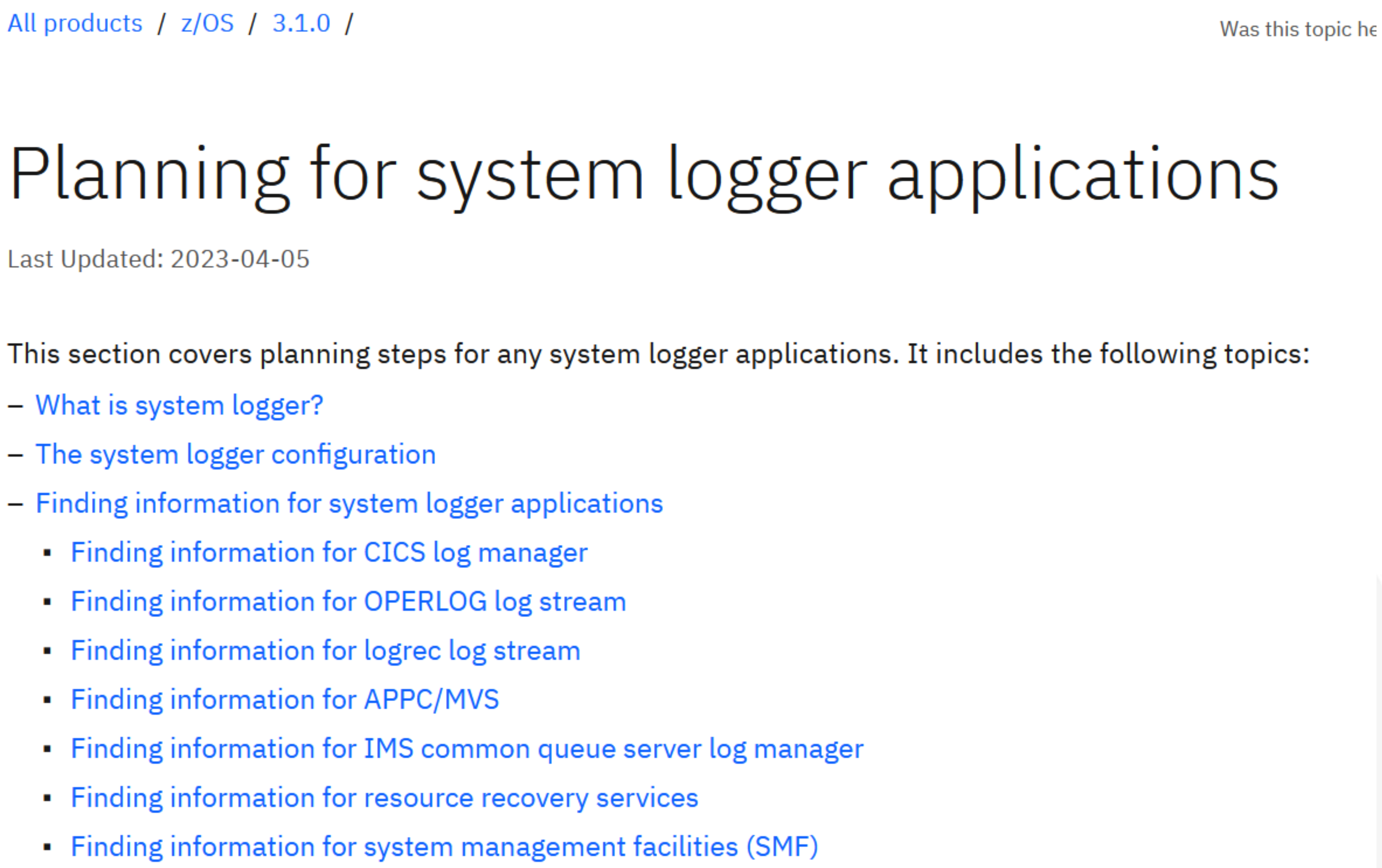


```
DSLIST - Data Sets Matching IXGLOGR.IFASMF.PLX3.DEFAULT.SND*          Row 1 of 4
Command ===> _                                                Scroll ===> CSR

Command - Enter "/" to select action                  Message           Volume
-------------------------------------------------------------------------------
         IXGLOGR.IFASMF.PLX3.DEFAULT.SNDA                               *VSAM*
         IXGLOGR.IFASMF.PLX3.DEFAULT.SNDA.DATA                          ESY300
         IXGLOGR.IFASMF.PLX3.DEFAULT.SNDB                               *VSAM*
         IXGLOGR.IFASMF.PLX3.DEFAULT.SNDB.DATA                          ESY300
***************************** End of Data Set list ****************************
```

LOGSTREAM USERS

Planning



All products / z/OS / 3.1.0 /

Was this topic he

Planning for system logger applications

Last Updated: 2023-04-05

This section covers planning steps for any system logger applications. It includes the following topics:

- What is system logger?
- The system logger configuration
- Finding information for system logger applications
  - Finding information for CICS log manager
  - Finding information for OPERLOG log stream
  - Finding information for logrec log stream
  - Finding information for APPC/MVS
  - Finding information for IMS common queue server log manager
  - Finding information for resource recovery services
  - Finding information for system management facilities (SMF)

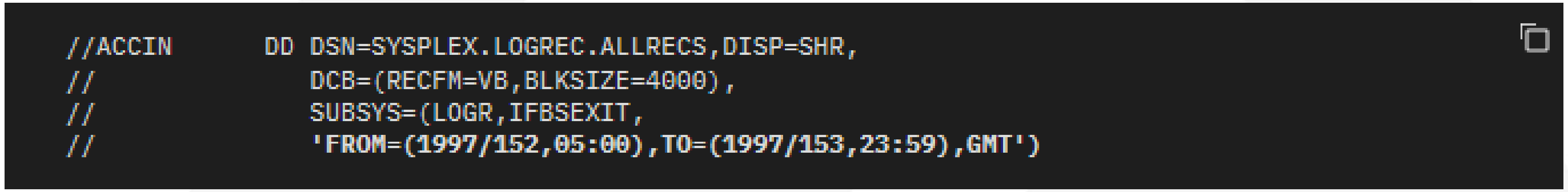
LINK

LOGREC

- Historically, LOGREC records were written to a sequential dataset
- To switch to logstream, in PARMLIB(IEASYSXX) specify

```
LOGREC=LOGSTREAM=LOGREC.TEST1
```

- IEASYS
- For a dynamic change use the SETLOGRC command
- For EREP reporting use the ACCIN DD card with the SUBSYS PARM e.g:



```
//ACCIN      DD DSN=SYSPLEX.LOGREC.ALLRECS,DISP=SHR,
//              DCB=(RECFM=VB,BLKSIZE=4000),
//              SUBSYS=(LOGR,IFBSEXIT,
//              'FROM=(1997/152,05:00),TO=(1997/153,23:59),GMT')
```

- EREP with LOGREC

OPERLOG

- Historically, the system log was written to SPOOL datasets(per LPAR) aka SYSLOG
- OPERLOG allows
  - SYSPLEX wide logging(not LPAR specific)
  - Coloring of messages like they appear on the console
  - With SDSF, allows log filtering on various fields
- Setup is easy:
  1. Define structure(for multisys) and logstream
  2. Issue this command V OPERLOG,HARDCPY
  3. Switch between SYSLOG and OPERLOG in SDSF with ‘LOG S’ and ‘LOG O’ respectively
  4. SDSF default is in ISFPRMxx
  5. S

Read all about it here

SMF

- Historically, SMF records were written to MAN datasets
- Using logstreams allows for:
  - SYSPLEX wide collection (not LPAR specific)
  - Automatic housekeeping using RETPD and AUTODELETE
  - Dumping utility allows for relative date filters(very handy!)
  - Separation of different SMF types to different logstreams
- Setup is not trivial :
  1. Plan how many logstreams, which record types will go to each logstream, retention periods
  2. Define structures(for multisys) and logstreams
  3. IN SMFPRMxx set RECORDING to LOGSTREAM
  4. In SMFPRM xx define as many LSNAME keywords as logstreams required
  5. T SMF=xx
  6. The hardest part is to migrate the existing SMF collection and reporting.
  7. Redbook

CICS

- DFHLOG and DFHSHUNT are the CICS system logs.
- They provide for the CICS system log, forward recovery logs, autojournals, and user journals.
- With a high transaction rate, tuning is crucial
- More information here

MISCELLANEA

Record deletion

- If records were never deleted from logstreams, you would run out of DASD space unless you hit the extent limit(168 is the default) first.
- Two ways of deleting records
  - Use AUTODELETE and RETPD. The logger will automatically delete any records older (in days) specified on the RETPD parameter.
  - Use the logstream owner’s utility if available:
    - SMF – use the ARCHIVE option with IFASMFDL
    - OPERLOG – use the IEAMDBLG program
    - LOGREC – use EREP program with the DELETE option
- Offload datasets are marked for deletion when all the records have been deleted. They are physically deleted the next time a new offload dataset is allocated.
- DO NOT delete the logstream offload datasets (such as TSO DELETE)

LOGRY and LOGRZ policies

- Problem
  - In GDPS environments, the k-Systems must be as isolated from the rest of the sysplex as possible to ensure that GDPS can accomplish failover
  - So in some GDPS configurations, z/OS System Logger services are not available on the k-Systems:
    - Logger address space not started at all, or gets cancelled by GDPS if it does start
    - Logger CDS not accessible (either not online or ALLOWACCESS=NO specified in IXGCNFxx)
  - But logging is very useful, even on the k-Systems

- Solution
  - Logger and XCF now support two new types of logger single system-scope couple data sets (CDS) that do not interfere with GDPS ability to accomplish failover
    - LOGRY and LOGRZ

- Benefit
  - The k-Systems can make use of system logger services
  - You can extract log data using same tools/utilities as other systems in the sysplex

With thanks to Steve Warren at IBM

Questions? Comments? Ideas?

www.share.org/evaluations