

---

# Analyzing CICS TS DFH0STAT Reports Tuning Your System

---

Gregory Rodriguez  
C\TREK Corporation  
PO Box 560069  
Montverde FL 34756-0069  
grodriguez@ctrek.com  
407-469-3600

---

# DISCLAIMERS/TRADemarks

- YMMV
- Remember the Political Factor
- CICS/VS, CICS/MVS, CICS/ESA, CICS TS, COBOL LE, COBOL 2, VSAM, DB2, OS/390, MVS, z/OS and z/VSE are Trademarks of the International Business Machines Armonk, NY

---

# Agenda

- Acknowledgement
- CICS Tools
- What is DFH\$STAT?
- Basic DFH0STAT Reports
  - System Status – Transaction Manager
  - Dispatcher – Storage Manager – Loader
  - Transactions – Programs – TS/TD
  - LSR – Files – Data Tables
- Closing

---

# Special Thank You!!

- We appreciate receiving data from several clients that have made this presentation more valuable
- In particular, we are grateful for the assistance in obtaining the different DFH0STAT reports and review and recommendations of this presentation to Mr. Karl DeVore

---

# CICS Tools

- CICS TS provides a series of tools with the basic CICS package that can be used to tune your system
  - End of Day Statistics (EOD)
    - Can be collected at intervals
    - Can be quite bulky
  - STAT Transaction
    - Sometimes called the “poor mans performance tool”
  - CICS supplied transactions
    - For example, CEMT
      - Display/Alter resources and other functions

---

# What is DFH\$STAT?

- DFH\$STAT is a CEDDA group that contains a series of resources that can be used to print out a CICS statistics report on demand
  - Transaction = STAT
  - BMS map = DFH0STM
  - Programs = DFH0STAT, DFH\$STAS, DFH\$STCN and DFH\$STED
- Issue CEDDA INS G(DFH\$STAT)
- Reports are usually spooled to POWER
  - Must include SPOOL SIT parameter set to “YES”

---

# DFH0STAT Reports

- Information regarding DFH0STAT can be found in the CICS TS Performance Guide Appendix B
  - Not very informative
- Topics covered by DFH0STAT
  - System Status, Monitoring and Statistics
  - Transaction Manager and Dispatcher
  - Storage
  - Loader
  - Transactions
  - Transaction Totals
  - Programs
  - Program Totals
  - Temporary Storage
  - Transient Data
  - LSR Pools
  - Files
  - Data Tables

# Report Creation

- Enter **STAT** to invoke DFH0STAT

```
DALLAS01 - EXTRAI X frame
Sample Program - CICS Statistics Print          02/04/10  10:45:23

Type in destination fields if required. Press Enter to print

Jobname . . . : CICSTK
Applid . . . : PRODCICS
Sysid . . . : CIC2

Node . . . : *                               Type in a valid Node. * is default
Userid . . . : *                             Type in a valid Userid. * is default
Class . . . : A                               Type in a valid Class. A is default

TS Queue Name _____ Type in TS Queue name, to send out-
                          put to this TS queue instead.
Abbreviated _____ Enter x for abbreviated TS report

F3=Exit to CICS

[4] [B] [00.1] [11/21]
```

Press **“ENTER”** to create the report



# DFH0STAT Output

## System Status, Monitoring and Statistics

■ Applid PRODCICS Sysid CIC2 Jobname CICSTK Date 02/04/10 Time 11:20:14 CICS 01.01.01 PAGE 1

### System Status

- VSE Release . . . . . : VSE/AF8.1.0
- CICS Startup . . . . . : COLD
- CICS Status . . . . . : ACTIVE
- Storage Protection . . . : ACTIVE
- Reentrant Programs . . . : PROTECT

### Monitoring

- Monitoring . . . . . : OFF
- Exception Class . . . : OFF
- Performance Class . . : OFF
- Exception Class Records . . . . . : 0
- Exception Records Suppressed . . . : 0
- Performance Class Records . . . . . : 0
- Performance Records Suppressed . . : 0
- DMF Records . . . . . : 0
- DMF Errors . . . . . : 0

### Statistics

- Statistics End-of-Day Time . . . : 00:00:00
- Statistics Interval . . . . . : 03:00:00
- Next Statistics Collection . . . : 00:00:00
- Statistics Recording . . . . . : OFF

Provides general information on how your CICS system was initialized, identifies the status of certain protection parameters and Monitoring information

Provides information regarding the Statistics Collection Interval. The collection interval, specifically, the Next Statistics Collection interval is important because it identifies how much data you have to analyze

# DFH0STAT Output

## System Status, Monitoring and Statistics

- Statistics
- \_\_\_\_\_
- Statistics End-of-Day Time . . . : 00:00:00
- Statistics Interval . . . . . : 03:00:00 ← **Interval**
- Next Statistics Collection . . . : 15:00:00 ← **Next Collection Time**
- Statistics Recording . . . . . : ON

15:00:00 Next  
-03:00:00 Interval  
12:00:00 Last Collection

- Applid PRODCICS Sysid CIC2 Jobname CICSPROD Date 01/27/10 Time 12:05:02 CICS 01.01.01

- Transaction Manager
- \_\_\_\_\_
- Total Accumulated transactions so far . . . : 161,917
- Accumulated transactions (since reset) . . . : 334 ← **Not much data accumulated in a 5 minute period**
- Maximum transactions allowed (MXT) . . . : 75
- Times at MXT . . . . . : 0
- Current Active User transactions . . . . . : 12
- Peak Active User transactions . . . . . : 15
- Total Active User transactions . . . . . : 332

# DFH0STAT Output

## Transaction Manager and Dispatcher

### Transaction Manager

- Total Accumulated transactions so far . . : 942,523
- Accumulated transactions (since reset) . . : 874,475
- Maximum transactions allowed (MXT) . . . : 85
- Times at MXT . . . . . : 2
- Current Active User transactions . . . . . : 14
- Peak Active User transactions . . . . . : 85
- Total Active User transactions . . . . . : 867,350
- Current Running transactions . . . . . : 1
- Current Dispatchable transactions . . . . . : 0
- Current Suspended transactions . . . . . : 13
- Current System transactions . . . . . : 0
- Transactions Delayed by MXT . . . . . : 420
- Total MXT queueing time . . . . . : 00:24:52.52408
- Average MXT queueing time . . . . . : 00:00:03.55361
- Current Queued User transactions . . . . . : 0
- Total Queueing time for current queued . . : 00:00:00.00000
- Average Queueing time for current queued : 00:00:00.00000

### Dispatcher

- Dispatcher start time . . . : 04:42:42.28621
- Peak tasks . . . . . : 99
- Current tasks . . . . . : 41
- Current ICV time . . . . . : 1,000ms
- Current ICVR time . . . . . : 20,000ms
- Current ICVTSD time . . . . : 0ms
- Current PRTYAGING time . . . : 5,000ms
- Number of active CICS TCBS : 4

**MXT Objective**

**MXT Limit Exceeded**

**Identifies possible MXT conditions that may need attention**

**Initialization settings**

---

## DFH0STAT Output

### Transaction Manager and Dispatcher

- Be careful with the ICVR and MXT settings
  - ICVR = 20000 or 20 seconds
  - MXT = 85
  - If transaction rate is:
    - At 5 tasks/second, then when a transaction is cancelled because of AICA, you will have 100 task in the queue (5 tasks/second X 20 seconds)
    - ICVR should be set at less than 1000 system wide
    - Do not use a high ICVR for a blanket OK for every task if only one or a few need more CPU cycles
      - Remember, the ICVR is reset with every task dispatch
      - Use CEDA DEFINE TRAN parameter RUNAWAY for these

# DFH0STAT Output

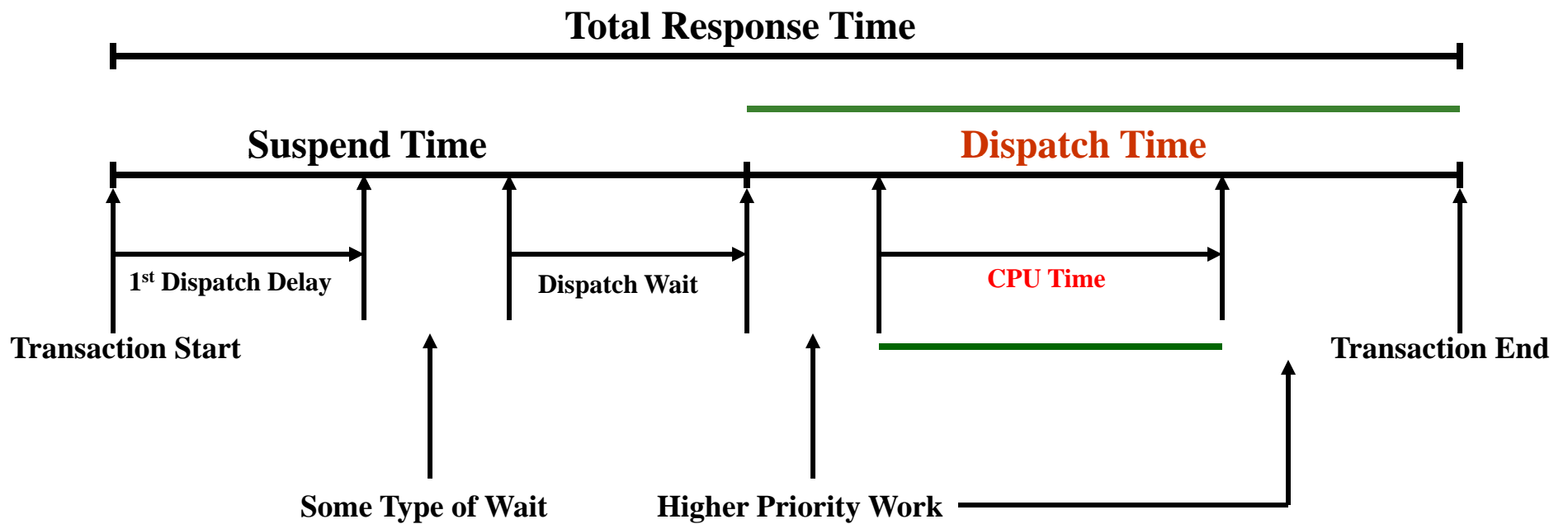
## Transaction Manager and Dispatcher

### Important Fields

TCB	TCB	TCB	Op. System	Op. System	TCB	TCB	DS TCB
Name	Status	Start Time	Waits	Wait Time	Dispatch Time	CPU Time	CPU Time
QR_SUBD	Active	04:42:42.28621	15,058,870	21:37:28.68838	01:25:15.79488	00:32:26.62400	00:00:00.00000
RO_SUBD	Active	04:42:42.72693	15,593	23:02:18.07875	00:00:26.40452	00:00:01.53600	00:00:00.00000
SO_MODE	Active	04:42:48.96055	31,440	23:02:33.79144	00:00:10.69181	00:00:01.02400	00:00:00.00000
SL_MODE	Active	04:42:48.96030	14,364	04:42:56.68896	00:00:07.72865	00:00:00.00000	00:00:00.00000
<b>Totals</b>					<b>01:26:00.61988</b>	<b>00:32:29.18400</b>	<b>00:00:00.00000</b>

- Important measurement statistic is called the “CPU to Dispatch Ratio”
  - Applies **only** to the QR TCB
- Operating System Waits – how many times the system has been in a no work to dispatch and issues an operating system wait (SVC 7)
  - Issues a Multiple Wait – waits on any outstanding I/O, VTAM RPL, ICV setting etc.
    - Any pending interrupt will activate CICS even if the ICV has **not** been completed
    - Affects CPU availability for lower priority partitions
      - Can represent wasted resources in low activity system

# Response Time



---

# DFH0STAT Output

## Transaction Manager and Dispatcher

- **TCB Dispatch Time** represents the time from when CICS hands control to a transaction to execute until the moment CICS receives control back from the transaction
  - Receives control voluntarily
    - EXEC CICS issued
      - Note: not all EXEC CICS return to the CICS Dispatcher
  - Receives control involuntarily
    - Transaction ABEND
    - ICVR – AICA
  - As far as CICS is concerned – Dispatch time is time the transaction is using the CPU
- **TCB CPU Time** – the actual CPU time used by the transaction
  - Transaction can lose control
    - Higher priority work
    - Paging
    - Operating System requests
    - Extrapartition requests
    - Hypervisors (z/VM or LPAR Management)

---

# DFH0STAT Output

## Transaction Manager and Dispatcher

- CPU to Dispatch Ratio
  - Ratio = ((TCB CPU Time/TCB Dispatch Time)\*100)
- Objective – 80% or better
  - Hard to achieve in a Hypervisor environment
    - Usually in the 30 – 60% range
- Current Example
  - TCB Dispatch Time = 01:25:15.79488 (5116 seconds)
  - TCB CPU Time = 00:32:26.62400 (1947 seconds)
  - Ratio = ((1947/5116)\*100) = 38%

---

## DFH0STAT Output

### Transaction Manager and Dispatcher

- CPU to Dispatch Ratio Formula (QR)

**Ratio = ((CPU Time / Dispatch Time) \* 100)**

**Dispatch Time = 01:25:15.79488 → 5,116 seconds**

**CPU Time = 00:32:26.62400 → 1,947 seconds**

**Ratio = ((1947 / 5116) \* 100 = 38%**

---

# DFH0STAT Output

## Transaction Manager and Dispatcher

- Some things that can affect CPU/Dispatch Ratio
  - LPAR Weights
    - Number of LPARs versus physical CPUs
    - Capping
  - z/VM
    - Share allocations
    - # of active guests
  - z/VSE Partition Priority
  - Paging
  - TD Extrapartition use
  - Use of operating system macros that can cause waits

---

# DFH0STAT Output

## Transaction Manager and Dispatcher

### ■ CPU to Dispatch Ratio

	Number of Phys CPUs	Number of LPARs	CPU Capping	# of z/VM Guests
User-A	1	1	No	03
User-B	1	3	No	00
User-C	1	1	No	00
User-D	3	9	No	00
User-E	2	1	No	14
User-F	1	1	No	03

**Note 1: User-A and User-F have NOPDS**

**Note 2: User-B has 1 IFL (LINUX) and 1 Inactive CP (Others may have additional CPs)**

---

---

# DFH0STAT Output

## Transaction Manager and Dispatcher

### ■ CPU to Dispatch Ratio


	Operating Environment	Total MIPS	CPU/Dispatch Ratio
User-A	z/VM and z/VSE	130	38%
User-B	Native z/VSE	87	39%
User-C	Native z/VSE	526	35%
User-D	Native z/VSE	783	43%
User-E	z/VM and z/VSE	472	58%
User-F	z/VM and z/VSE	110	34%

# DFH0STAT Output

## Storage Manager

- Partition size established from ALLOC parameter . . . : 265,727K
- Storage BELOW 16MB
- 
- ---
- Partition GETVIS area size under 16 Mb . . . . . : 9,724K
- Partition GETVIS used area below 16 Mb . . . . . : 8,656K
- Partition GETVIS free area below 16 Mb . . . . . : 1,068K
- Partition GETVIS maximum used below 16 Mb . . . . . : 9,724K
- Partition GETVIS largest free area below 16 Mb . . : 1,064K
- ---
- Current DSA Limit . . . . . : 6,144K
- Current Allocation for DSAs . . : 6,144K
- Peak Allocation for DSAs . . . : 6,144K

→ **Important figure as it represents how much space you have left to expand below the line**

 → **Provides an indication of how well your DSALIM parameter has been set and can serve as a possible warning for potential SOS conditions**

**Information for EDSALIM is similar and will not be covered**

# DFH0STAT Output

## Storage Manager

	CDSA	UDSA	SDSA	RDSA	Totals	
-						
+						
Current DSA Size . . . . .	1,792K	3,328K	512K	512K	6,144K	
Current DSA Used . . . . .	1,676K	304K	424K	500K	2,904K	
Current DSA Used as % of DSA . .	93%	9%	82%	97%	47% of DSA Size	→ % can be misleading
* Peak DSA Used . . . . .	1,676K	3,328K	440K	500K		
Peak DSA Size . . . . .	2,048K	3,328K	512K	512K		
Cushion Size . . . . .	64K	64K	64K	64K		
Free Storage (inc. Cushion) . .	116K	3,024K	88K	12K		
* Peak Free Storage . . . . .	1,240K	3,064K	396K	64K		
* Lowest Free Storage . . . . .	116K	0K	72K	12K		
Largest Free Area . . . . .	32K	256K	88K	12K		
Largest Free Area as % of DSA :	1%	7%	17%	2%		
Largest Free/Free Storage . . .	0.27	0.08	1.00	1.00		
Current number of extents . . .	7	13	2	2	24	
Number of extents added . . . .	3	8	0	0		
Number of extents released . . .	1	0	0	0		
Getmain Requests . . . . .	132,482	40,868,521	1,242	1		
Freemain Requests . . . . .	132,010	40,868,520	1,210	3		
Current number of Subpools . . .	48	23	6	4	81	
Add Subpool Requests . . . . .	874,616	874,616	0	0		
Delete Subpool Requests . . . .	874,617	874,617	0	0		
Times no storage returned . . . .	0	4	0	0		
Times request suspended . . . .	0	3,223	0	0		→ SOS indicators
Current requests suspended . . .	0	0	0	0		
Peak requests suspended . . . .	0	44	0	0		
Requests purged while waiting :	0	41	0	0		
Times Cushion released . . . . .	0	648	0	0		
Times Short-On-Storage . . . . .	0	23	0	0		
Total time Short-On-Storage . .	00:00:00.00000	00:07:52.07785	00:00:00.00000	00:00:00.00000		
Average Short-On-Storage time :	00:00:00.00000	00:00:20.52513	00:00:00.00000	00:00:00.00000		→ SV indicator
Storage Violations . . . . .	0	0	0	0	0	
Access . . . . .	CICS	USER	USER	READONLY		
** indicates values reset on last DSA Size change						

# DFH0STAT Output

## Storage Manager

- Current DSA Limit . . . . . : **5,120K** 1
- Current Allocation for DSAs . . . . . : 2,816K
- Peak Allocation for DSAs. . . . . : 2,816K

	CDSA	UDSA	SDSA	RDSA	Totals	
Current DSA Size. . . . . :	1,280K	256K	768K	512K	<b>2,816K</b>	<b>22</b>
Current DSA Used. . . . . :	968K	12K	648K	432K	2,060K	
Current DSA Used as % of DSA. :	75%	4%	84%	84%	73% of DSA Size	

Note: DSALIM specified

% is based on allocated amount and not on DSALIM

**A high % in this column (e.g., 95%) does not necessarily mean that you are close to an SOS condition – If Current DSA Limit (1) is greater than Current Total DSA Size (2), then you have room to expand the DSAs by adding an additional extent**

# DFH0STAT Output

## Storage Manager

- Storage ABOVE 16MB

- Partition GETVIS area size above 16 Mb . . . . . : 215,032K
- Partition GETVIS used area above 16 Mb . . . . . : 204,500K
- Partition GETVIS free area above 16Mb . . . . . : 10,532K
- Partition GETVIS maximum used above 16 Mb . . . . . : 205,548K
- Partition GETVIS largest free area above 16 Mb . . : 10,440K

- Current EDSA Limit. . . . . : 179,200K
- CICS Trace table size . . . . . : 4,096K
- Current Allocation for EDSAs. : 49,152K
- Peak Allocation for EDSAs . . : 49,152K

	ECDSA	EUDSA	ESDSA	ERDSA	Totals
Current DSA Size. . . . . :	14,336K	5,120K	24,576K	5,120K	49,152K
Current DSA Used. . . . . :	12,476K	64K	23,724K	4,364K	40,628K
Current DSA Used as % of DSA. : 87%		1%	96%	85%	<b>82% of EDSA</b>
<b>Size</b>					
* Peak DSA Used . . . . . :	13,444K	4,736K	23,732K	4,364K	
Peak DSA Size . . . . . :	14,336K	5,120K	24,576K	5,120K	
Cushion Size. . . . . :	128K	0K	128K	256K	

A high Used % but insignificant

# DFH0STAT Output

## Loader

■	Loader				
■	Library Load requests. . . . .	2,123	Total Program Uses . . . . .	13,209,034	
■	Total Library Load time. . . . .	00:00:17.48836	Program Use to Load Ratio. . . . .	221.87	← Ratio is incorrect due to truncation
■	Average Library Load time. . . . .	00:00:00.00822			
■	Library Load requests that waited. . . . .	0			
■	Total Library Load request wait time . . . . .	00:00:00.00000			13209034/2123=6221.87
■	Average Library Load request wait time . . . . .	00:00:00.00000			
■	Current Waiting Library Load requests. . . . .	0			
■	Peak Waiting Library Load requests . . . . .	0			
■	Times at Peak. . . . .	0	Average Not-In-Use program size. . . . .	28K	

### Miscellaneous Load Data

**Average Load Time is around 8.22 ms.**

**Not In Use (NIU) Program – a program that is no longer in use and can be compressed out of storage**

# DFH0STAT Output

## Loader

■ CDSA		ECDSA	
■ <u>Programs Removed by compression</u> . . . . .	704	Programs Removed by compression. . . . .	0
■ Time on the Not-In-Use Queue . . . . .	18:32:34.25712	Time on the Not-In-Use Queue . . . . .	00:00:00.00000
■ Average Time on the Not-In-Use Queue . . . . .	00:00:43.99913	Average Time on the Not-In-Use Queue . . . . .	00:00:00.00000
■ Programs Reclaimed from the Not-In-Use Queue . . . . .	171,136	Programs Reclaimed from the Not-In-Use Queue . . . . .	1,561,905
■ Programs Loaded - now on the Not-In-Use Queue . . . . .	299	Programs Loaded - now on the Not-In-Use Queue . . . . .	84
■ SDSA		ESDSA	
■ <u>Programs Removed by compression</u> . . . . .	1	Programs Removed by compression. . . . .	0
■ Time on the Not-In-Use Queue . . . . .	01:34:16.15213	Time on the Not-In-Use Queue . . . . .	00:00:00.00000
■ Average Time on the Not-In-Use Queue . . . . .	00:00:50.38662	Average Time on the Not-In-Use Queue . . . . .	00:00:00.00000
■ Programs Reclaimed from the Not-In-Use Queue . . . . .	579,200	Programs Reclaimed from the Not-In-Use Queue . . . . .	6,826,026
■ Programs Loaded - now on the Not-In-Use Queue . . . . .	68	Programs Loaded - now on the Not-In-Use Queue . . . . .	1,519
■ RDSA		ERDSA	
■ <u>Programs Removed by compression</u> . . . . .	3	Programs Removed by compression. . . . .	0
■ Time on the Not-In-Use Queue . . . . .	21:35:51.39046	Time on the Not-In-Use Queue . . . . .	00:00:00.00000
■ Average Time on the Not-In-Use Queue . . . . .	02:04:01.30004	Average Time on the Not-In-Use Queue . . . . .	00:00:00.00000
■ Programs Reclaimed from the Not-In-Use Queue . . . . .	0	Programs Reclaimed from the Not-In-Use Queue . . . . .	130,435
■ Programs Loaded - now on the Not-In-Use Queue . . . . .	0	Programs Loaded - now on the Not-In-Use Queue . . . . .	43

**These are the (E) DSAs that can have programs**

**The (E) UDSA contains no executable programs in it**

**Program compression is your Early Warning System that you may be approaching SOS**

# DFH0STAT Output

## Loader

- Program Storage
- 
- |   |        |  |         |
|---|--------|--|---------|
| Nucleus Program Storage (CDSA) . . . . .            | 1,136K | Nucleus Program Storage (ECDSA) . . . . .            | 3,044K  |
| Program Storage (SDSA) . . . . .                    | 320K   | Program Storage (ESDSA) . . . . .                    | 54,040K |
| Resident Program Storage (SDSA) . . . . .           | 4K     | Resident Program Storage (ESDSA) . . . . .           | 696K    |
| Read-Only Nucleus Program Storage (RDSA) . . . . .  | 60K    | Read-Only Nucleus Program Storage (ERDSA) . . . . .  | 1,320K  |
| Read-Only Program Storage (RDSA) . . . . .          | 116K   | Read-Only Program Storage (ERDSA) . . . . .          | 2,168K  |
| Read-Only Resident Program Storage (RDSA) . . . . . | 44K    | Read-Only Resident Program Storage (ERDSA) . . . . . | 0K      |
- CDSA used by Not-In-Use programs : 1,091K 60.87% of CDSA ECDSA used by Not-In-Use programs : 2,052K 14.31% of ECDSA
- SDSA used by Not-In-Use programs : 298K 58.28% of SDSA ESDSA used by Not-In-Use programs : 53,212K 92.80% of ESDSA
- RDSA used by Not-In-Use programs : 52K 10.15% of RDSA ERDSA used by Not-In-Use programs : 582K 8.11% of ERDSA

**These statistics provide information as to how much (E) DSA storage is being used by programs. The NIU % relate how much program compression is possible in case we run short on storage**

# DFH0STAT Output

## Transaction

Tran id	Tran Class	Program Name	Task Data Dynamic	Task Data Location/Key	Attach Count	Restart Count	Dynamic Local	Dynamic Remote	Counts Remote Starts
MQBX		MQBICIRH	Static	Below/USER	1	0	0	0	0
MQER		MQPERR	Static	Below/USER	2,790	0	0	0	0
MQ01		MQPRECV	Static	Below/USER	7	0	0	0	0
MQ02		MQPAIP2	Static	Below/USER	397	0	0	0	0
MVAA		MVO0C005	Static	Any/USER	39	0	0	0	0
MV00		MVO0C000	Static	Any/USER	5	0	0	0	0
N		NOPROG	Static	Below/USER	1	0	0	0	0
NAM1		CLAMNAM	Static	Any/USER	192	0	0	0	0
NR01		GSCO8001	Static	Any/USER	121	0	0	0	0
OPNC		OPENCNT	Static	Below/USER	3,437	0	0	0	0
PA00		PAO0C000	Static	Any/USER	4,458	0	0	0	0
PA50		PAO0C005	Static	Any/USER	667,051	0	0	0	0

### Transaction Totals

Task Data Location/Key	Transaction Count	Attach Count
Below/CICS	32	7,900
Any/CICS	47	30,403
Below/USER	163	36,353
Any/USER	1,277	799,819
<b>Totals</b>	<b>1,519</b>	<b>874,475</b>

**This report is used to determine the location of the data and the protect key assigned**

**The summary is good for when you need to estimate an effort to convert programs to support above the line storage**

# DFH0STAT Output

## Program

Program Name	Data Loc	Exec Key	Times Used	Times Fetched	Total Fetch Time	Average Fetch Time	Times Newcopy	Times Removed	Program Size	Program Location
S1S610	Below	CICS	17,634	6	00:00:00.02323	00:00:00.00387	0	6	5,930	CDSA
S1S611	Below	CICS	9,352	26	00:00:00.01920	00:00:00.00073	0	26	1,474	CDSA
S1S615	Below	CICS	4,695	35	00:00:00.02601	00:00:00.00073	0	35	2,130	CDSA
S1S750	Below	CICS	4,492	4	00:00:00.00398	00:00:00.00099	0	4	2,074	CDSA
S1S885	Below	CICS	486	3	00:00:00.01625	00:00:00.00540	0	3	602	CDSA
S1S900	Below	CICS	428	4	00:00:00.00998	00:00:00.00249	0	4	594	CDSA
TBLM002			8	2	00:00:00.01585	00:00:00.00792	0	1	274	CDSA
TBLM010			16	1	00:00:00.00715	00:00:00.00715	0	0	6,017	ECDSA
USERLIST	Below	USER	1	1	00:00:00.00894	00:00:00.00894	0	0	11,866	ESDSA
UTLMS02			8	2	00:00:00.01188	00:00:00.00593	0	1	1,777	CDSA

If Times Fetched is greater than 1, then this may be an indication that your (E) DSALIM is too small but must check Times Newcopy to ensure fetches were not due to other activity

Indication of program compression

# DFH0STAT Output

## Program

### ■ Program Totals

■	Programs.....:	6,435
■	Assembler.....:	1,940
■	C.....:	18
■	COBOL.....:	4,457
■	LE/VSE.....:	0
■	PL1.....:	5
■	Other.....:	15
■	Maps.....:	1,362
■	Partitionsets.....:	1
■	<hr/>	
■	Total.....:	7,798
■	CDSA Programs.....:	306
■	SDSA Programs.....:	72
■	RDSA Programs.....:	5
■	ECDSA Programs.....:	107
■	ESDSA Programs.....:	1,543
■	ERDSA Programs.....:	72
■	SVA Programs.....:	0
■	ESVA Programs.....:	0
■	Unused Programs.....:	73
■	Not Located Programs.....:	5,620
■	<hr/>	
■	Total.....:	7,798

**General totals  
regarding programs  
defined to the system**

**Many non-reentrant programs  
Exposure: Hard to debug SV  
in (S) DSAs because you  
don't get the areas dumped**



# DFH0STAT Output

## Temporary Storage

■ Temporary Storage _____			
■ Put/Putq main storage requests . . . . .	1,730		
■ Get/Getq main storage requests . . . . .	182,701		
■ Peak storage used for TS Main. . . . .	200K		
■ Current storage used for TS Main . . . . .	21K		
■ Put/Putq auxiliary storage requests. . . . .	1,867,565	← Potential I/O	
■ Get/Getq auxiliary storage requests. . . . .	2,152,355		
■ Times temporary storage queue created. . . . .	247,080		
■ Peak temporary storage queues in use . . . . .	2,782		
■ Current temporary storage queues in use. . . . .	2,734		
■ Items in longest queue . . . . .	9,121		
■ Queue extension threshold. . . . .	20		
■ Queue extensions created . . . . .	33,043		
■ Control interval size. . . . .	4,096		
■ Control intervals in the DFHTEMP dataset :	26,342	← Used for sizing DFHTEMP	
■ Peak control intervals used. . . . .	22,003		
■ Current control intervals in use . . . . .	20,506		
■ Available bytes per control interval . . . . .	4,032		
■ Segments per control interval. . . . .	63		
■ Bytes per segment . . . . .	64		
■ Writes bigger than control interval size :	39,450	← Can affect TS performance-affects CISZ	
■ Largest record length written. . . . .	32,795		
■ Times auxiliary storage exhausted. . . . .	0	← Should not occur	
■ Number Temporary storage compressions. . . . .	955,399	← Result from reusing CIs	
■ Temporary storage strings. . . . .	8		
■ Peak Temporary storage strings in use. . . . .	8		
■ Temporary storage string waits . . . . .	25		
■ Peak users waiting on string . . . . .	8		
■ Current users waiting on string. . . . .	0	→ Wait conditions that should be addressed	
■ Temporary storage buffers. . . . .	52		
■ Temporary storage buffer waits . . . . .	1,439		
■ Peak users waiting on buffer . . . . .	6		
■ Current users waiting on buffer. . . . .	0		
■ Temporary storage buffer reads . . . . .	183,152	← Physical I/O	
■ Temporary storage buffer writes. . . . .	361,387		
■ Forced buffer writes for recovery. . . . .	0		
■ Format writes. . . . .	0	← New extent added	
■ I/O errors on the DFHTEMP dataset. . . . .	0	← Don't want to see this!!	

---

# DFH0STAT Output

## Temporary Storage

### ■ Reduce I/O – Simulate TS Main

□ Put/Putq auxiliary storage requests. . . :	1,867,565
□ Get/Getq auxiliary storage requests. . . :	2,152,355
□ Temporary storage buffer reads . . . . . :	183,152
□ Temporary storage buffer writes. . . . . :	361,387

### ■ Objective is LT 20% Physical I/O Ratio

□ 
$$\text{Ratio} = \frac{(\text{TS Buffer Reads} + \text{TS Buffer Writes}) * 100}{(\text{PUTQ AUX} + \text{GETQ AUX})}$$

**Ratio = ((183152+361387) / (1867565+2152355)) \* 100**

**Ratio = (544539 / 4019920) \* 100 = 14% (Physical I/O Operations)**

**Look Aside Hit Ratio = 100% - 14% = 86%**

---

## DFH0STAT Output Temporary Storage

- Improve look-aside hit ratio or reduce I/O by adding DFHTEMP buffers
  - Buffers are not written to DFHTEMP until needed
    - So, more buffers allow the data to sit in storage providing an opportunity for a task to re-acquire the data in the buffer
    - If the data is reused and deleted before the buffer is required, then no I/O takes place
    - Simulates TS Main
    - As the buffers are reused, expect an increase in TS compressions

---

## DFH0STAT Output

### Temporary Storage

- Use peak TS CIs used to see if DFHTEMP needs to be expanded
  - Maximum CIs in DFHTEMP
    - Secondary extents
    - Maximum of 64K CIs minus 1
- Writes GT CISZ causes additional overhead and is a major contributor to buffer and string waits
  - Avoid using largest record written as a base because it is a HWM which does not necessarily reflect the remaining records that were GT than the CISZ
  - Pick a CISZ that will eliminate at least 75% of the writes GT CISZ

# DFH0STAT Output

## Transient Data

■	Transient Data		
■	-----		
■	Transient data reads . . . . .	0	
■	Transient data writes. . . . .	27	
■	Transient data formatting writes . . . . .	0	
■	Control interval size. . . . .	4,096	
■	Control intervals in the DFHNTRA dataset :	1,080	
■	Peak control intervals used. . . . .	4	
■	Times NOSPACE on DFHNTRA occurred. . . . .	0	← Should not occur
■	Transient data strings . . . . .	3	
■	Times Transient data string in use . . . . .	27	
■	Peak Transient data strings in use . . . . .	2	
■	Times string wait occurred . . . . .	0	← Should not occur
■	Peak users waiting on string . . . . .	0	
■	Transient data buffers . . . . .	3	
■	Times Transient data buffer in use . . . . .	13,253	
■	Peak Transient data buffers in use . . . . .	2	
■	Peak buffers containing valid data . . . . .	3	
■	Times buffer wait occurred . . . . .	0	← Should not occur
■	Peak users waiting on buffer . . . . .	0	
■	I/O errors on the DFHNTRA dataset. . . . .	0	← Should not occur

DFHINTRA tuning is similar to DFHTEMP

Avoid Extrapartition data sets

# DFH0STAT Output

## LSR

### LSR Pools

Pool Number : 1 Time Created : 04:43:01.17367

Maximum key length . . . . . : 125  
 Total number of strings . . . . : 14  
 Peak concurrently active strings : 12  
 Total requests waited for string : 0  
 Peak requests waited for string. : 0

**Objective: Peak should be between 50 -70% of the Total Strings**

### Buffer Totals

Data Buffers . . . . . : 153  
 Successful look asides . . . . : 7,302,137  
 Buffer reads . . . . . : 520,695  
 User initiated writes. . . . . : 98,918  
 Non-user initiated writes. . . . : 0

Index Buffers. . . . . : 0  
 Successful look asides . . . . : 0  
 Buffer reads . . . . . : 0  
 User initiated writes. . . . . : 0  
 Non-user initiated writes. . . . : 0

**No index buffers assigned**

### Data and Index Buffer Statistics

Size	Buffers	Look Asides	Reads	User Writes	Writes
512	15	2,409,798	35,178	88	0
1024	15	673,537	6,505	29	0
2048	25	595,318	48,518	772	0
4096	20	1,789,346	240,519	26,939	0
8192	40	630,179	132,543	70,405	0
12288	8	1,174,254	39,655	71	0
16384	3	3,346	1,091	614	0
24576	27	26,359	16,686	0	0

$$HR = (\text{Look-Asides} / (\text{Look-Asides} + \text{Reads})) * 100$$

$$HR = (7302137 / (7302137 + 520695)) * 100$$

$$HR = (7302137 / 7822832) * 100 = 93\%$$

**Suggested Objectives:**

**Data HR → 80%+**

**Index HR → 95%+**

**Combined HR → 93%+**

**Perform HR calculation by buffer**

# DFH0STAT Output

## LSR

### LSR Pools

Pool Number : 7 Time Created : 04:43:01.91474

Maximum key length . . . . .	100
Total number of strings . . . . .	20
Peak concurrently active strings :	2
Total requests waited for string :	0
Peak requests waited for string. :	0

### Buffer Totals

Data Buffers . . . . .	22	Index Buffers. . . . .	0
Successful look asides . . . . .	88,921	Successful look asides . . . . .	0
Buffer reads . . . . .	21,114	Buffer reads . . . . .	0
User initiated writes. . . . .	2,589	User initiated writes. . . . .	0
Non-user initiated writes. . . . .	0	Non-user initiated writes. . . . .	0

### Data and Index Buffer Statistics

Size	Buffers	Look Asides	Reads	User Writes	Writes
4096	11	16,173	21,104	2,589	0
8192	11	72,748	10	0	0

$$HR = (88921 / (88921 + 21114)) * 100$$

$$HR = (88921 / 110035) * 100 = 81\%$$

# DFH0STAT Output

## File

Filename	Access Method	LSR Type	Pool	Str Max	Waits Total	Read Requests	Get Update Requests	Browse Requests	Add Requests	Update Requests	Delete Requests	Data EXCPs	Index EXCPs
EXDINDX	VSAM	KSDS	1	0	0	313,515	0	4,523	0	0	0	41,023	5,099
EXQHOLD	VSAM	KSDS	1	0	0	4,475	5,109	2,938	745	3,192	57	10,059	1,830
EXFCNFG	VSAM	KSDS	7	0	0	32,752	2,552	0	0	2,552	0	22,037	0
EXFI008	VSAM	KSDS	3	0	0	849,187	906	897	897	906	0	282,133	20,490
EXAINDX	VSAM	KSDS	2	2	4	1,226,305	37,520	6	689	36,918	0	56,178	17,302
EXAMSTR	VSAM	KSDS	2	0	0	3,479,372	27,006	5,520,015	119,739	16,716	115,878	232,033	14,717
EXSCTY	VSAM	KSDS	0	11	782	5,115,003	13,534	0	0	8,956	0	5,137,701	5,128,584

String waits

- 1) File level
- 2) Pool level

EXCPs represent physical I/O to the device

In the case of an LSR file, you would need to relate the Pool results with the EXCPs for the file even though the Hit Ratio (HR) may be very good.

In the case of an NSR file, you would need to evaluate the BUFNIs assigned to ensure that you have the proper Hit Ratio for the index portion

---

# DFH0STAT Output

## File

- For what do you want to look in the file statistics?
  - Look for a high EXCP count versus file requests issued
    - Sum all requests (Reads + Get for Update + Browse + Adds + Updates + Deletes)
    - NSR
      - Look at the data and index EXCPs
      - If file index buffering is right, you should have at worst case a index to data EXCP ratio of around 1.0 or less
    - LSR
      - Look at the data and index EXCPs
      - Ensure that the index and data buffers are providing the proper look-aside hit ratio

# DFH0STAT Output File

## ■ NSR Example

Access	LSR	Str	Waits	Read	Get	Update	Browse	Add	Update	Delete	Data	Index	
Filename	Method	Type	Pool	Max	Total	Requests	Requests	Requests	Requests	Requests	EXCPs	EXCPs	
EXDET	VSAM	KSDS	0	0	0	16	20,652	1,288,330	12,915	20,424	70	159,403	228,775

## ■ Calculate total CICS requests

- Sum = 1,342,407
- Data EXCP = 159,403
- Index EXCP = 228,775
- Data to Index EXCP Ratio
  - Ratio = (Index EXCP / Data EXCP)
  - Ratio = (228775 / 159403) = 1.44
- The probability is that BUFNI is not set correctly

# DFH0STAT Output

## File

- Let us review this particular file – EXFCNFG

- Important file information from statistics

- READS – 32,757
    - GET/UPDATE – 2,552
    - UPDATES – 2,552
    - DATA EXCP – 22,037
    - INDEX EXCP – 0

- Important information regarding LSR Pool 7

- Data Buffers . . . . . : 22                      Index Buffers. . . . . : 0
    - Successful look asides . . . . : 88,921                      Successful look asides . . . . : 0
    - Buffer reads . . . . . : 21,114                      Buffer reads . . . . . : 0
    - User initiated writes. . . . . : 2,589                      User initiated writes. . . . . : 0
    - Non-user initiated writes. . . : 0                      Non-user initiated writes. . . : 0

- Data and Index Buffer Statistics

	Size	Buffers	Look Asides	Reads	User Writes	Writes
	4096	11	16,173	21,104	2,589	0
	8192	11	72,748	10	0	0

- EXFCNFG VSAM KSDS 7 0 0 32,752 2,552 0 0 2,552 0 22,037 0

**EXCPs**

# DFH0STAT Output

## File

### ■ Important LISTCAT information

- ATTRIBUTES (Data)
  - KEYLEN-----100 AVGLRECL-----2048 BUFSPACE-----13312 **CISIZE-----4096**
  - RKP-----0 MAXLRECL-----2048 EXCPEXIT----- (NULL) CI/CA-----180
  - SHROPTNS(2,3) RECOVERY SUBALLOC NOERASE NOCOMPRESS INDEXED NOWRITECHK NOIMBED
  - NOREPLICAT UNORDERED NOREUSE NONSPANNED
- STATISTICS
  - **REC-TOTAL-----658** SPLITS-CI-----249 EXCPS-----3255229
  - REC-DELETED-----3 SPLITS-CA-----4 EXTENTS-----1
  - REC-INSERTED-----251 FREESPACE-%CI-----0 SYSTEM-TIMESTAMP:
  - REC-UPDATED-----249029 FREESPACE-%CA-----0 2010.037 02:55:29
  - REC-RETRIEVED----4631293 **FREESPACE-----2949120** X'C57F5FC29E738904'
- ALLOCATION
  - SPACE-TYPE-----CYLINDER
  - SPACE-PRI-----11 USECLASS-PRI-----0 **HALRBA-OR-CI----8110080**
  - SPACE-SEC-----3 USECLASS-SEC-----0 **HUSRBA-OR-CI----5160960**
- ATTRIBUTES (Index)
  - **KEYLEN-----100** AVGLRECL-----0 BUFSPACE-----0 **CISIZE-----5120**
  - RKP-----0 MAXLRECL-----5113 EXCPEXIT----- (NULL) CI/CA-----9
  - SHROPTNS(2,3) RECOVERY SUBALLOC NOERASE NOCOMPRESS NOWRITECHK NOIMBED NOREPLICAT
  - UNORDERED NOREUSE
- STATISTICS
  - **REC-TOTAL-----8** SPLITS-CI-----0 EXCPS-----4830 INDEX:
  - REC-DELETED-----0 SPLITS-CA-----0 EXTENTS-----1 **LEVELS-----2**
  - REC-INSERTED-----0 FREESPACE-%CI-----0 SYSTEM-TIMESTAMP: ENTRIES/SECT-----13

# DFH0STAT Output

## Data Tables

Data Tables - Requests		Maximum Table Size	Successful Reads	Records Not Found	Records in Table	Adds via Read	Adds via API	By Global User Exit	Adds Rejected	Rewrite Requests	Delete Requests	Read Retries
Filename	Table Type	Max Num recs										
IESFILE	CICS/TABLE	500	5,512	13	12	0	0	0	0	0	0	0

Data Tables - Storage		# of Requests to the Table		# of API Reads That Went to Source DS Because of N/F		Potential Physical I/O					
Filename	Type	Current Records	Peak Records	<----- Total -----> Storage Allocated	Storage In-Use	<----- Entries -----> Storage Allocated	Storage In-Use	<----- Index -----> Storage Allocated	Storage In-Use	<----- Data -----> Storage Allocated	Storage In-Use
IESFILE	CICS	12	12	192	3	32	1	32	1	128	1

Access		LSR	Str	Waits	Read	Get Update	Browse	Add	Update	Delete	Data	Index	
Filename	Method	Type	Pool	Max	Total	Requests	Requests	Requests	Requests	Requests	EXCPs	EXCPs	
IESFILE	VSAM	KSDS	1	0	0	0	0	26	0	0	0	2	2

**Storage Allocated/Used by the Data Table** (points to Storage Allocated/In-Use in the Storage table)  
**Usually to Load the Data Table** (points to Browse Requests in the Access table)  
**Reflects I/O to the Data Table for Non-Read Requests (After Load)** (points to Data and Index EXCPs in the Access table)

# DFH0STAT Output

## Data Tables

### Data Tables - Requests

Filename	Data Table Type	Max Num recs	Successful Reads	Records Not Found	Adds via Read	Adds via API	Adds Rejected	Rewrite Requests	Delete Requests	Read Retries
EXNAME	CICS/TABLE	350	80,888	0	294	0	0	0	0	0
EXAVAIL	CICS/TABLE	810,000	60,311,428	1	799,096	3,846	0	27,886	1,692	0
EXPOINT	CICS/TABLE	21,500	5,094,167	0	21,086	1	0	1	20	0
EXSITE	CICS/TABLE	6,500	46,629,826	0	4,866	0	0	0	0	0
EXMAST	CICS/TABLE	16	409,794	0	3	0	0	0	0	0
EXMSGs	CICS/TABLE	32	2,232,436	0	27	0	0	0	0	0

### Data Tables - Storage

Filename	Type	Current Records	Peak Records	<----- Total ----->		<----- Entries ----->		<----- Index ----->		<----- Data ----->	
				Storage Allocated	Storage In-Use	Storage Allocated	Storage In-Use	Storage Allocated	Storage In-Use	Storage Allocated	Storage In-Use
EXNAME	CICS	294	294	320	147	32	7	32	4	256	136
EXAVAIL	CICS	801,250	801,264	257,056	242,753	37,600	37,559	33,856	33,832	185,600	171,362
EXPOINT	CICS	21,067	21,086	15,424	14,819	672	659	544	519	14,208	13,641
EXSITE	CICS	4,866	4,866	19,808	13,092	128	115	96	65	19,584	12,912
EXMAST	CICS	3	3	192	4	32	1	32	1	128	2
EXMSGs	CICS	27	27	192	9	32	2	32	2	128	5

Filename	Access Method	Type	LSR Pool	Str Max	Waits Total	Read Requests	Get Update Requests	Browse Requests	Add Requests	Update Requests	Delete Requests	Data Index	
												EXCP	EXCPs
EXAVAIL	VSAM	KSDS	3	0	0	0	210,930	799,098	3,846	27,886	1,692	139,662	2,116
EXPOINT	VSAM	KSDS	1	0	0	0	1	21,087	1	1	20	2,139	41
EXSITE	VSAM	KSDS	2	0	0	0	0	4,867	0	0	0	609	26

---

# DFH0STAT Output

## Data Tables

- Lets Review EXAVAIL
  - Space requested = 810,000 records
  - Peak # of records = 801,264 or **98.9%**
    - Is it a problem? – “It depends”
      - Is this file growing?
      - ROT
        - 90% for static files
        - 80% for dynamic files
  - Storage
    - Allocated – 257 MB
    - Used – 243 MB
    - Utilized – 94.6%

---

# DFH0STAT Output

## Data Tables

- Lets Review EXAVAIL
  - Data Table Statistics
    - Read Only Requests – 60,311,428 (99.94%)
    - Non-Read Requests – 33,424
      - Adds – 3,846
      - Rewrites – 27,886
      - Deletes – 1,692
    - ROT – 90%+ Read Only

---

# DFH0STAT Output

## Data Tables

- Lets Review EXAVAIL
  - File Control Table (FCT) Statistics
    - Get for Update – **201,930**
    - Browse – 799,098
    - Adds – 3,486
    - Rewrites – 27,886
    - Deletes – 1,692
  - FCT EXCPs
    - Data – 139,662
    - Index – 2,116
  - Actual Read Only Ratio in the Data Table = 99.61%

# DFH0STAT Output

## Data Tables

### Files in LSR Pool 3

Access	LSR Str	Waits	Read	Get Update	Browse	Add	Update	Delete	Data	Index
Filename Method Type	Pool	Max Total	Requests	Requests	Requests	Requests	Requests	Requests	EXCPs	EXCPs
EXPEND VSAM KSDS	3	0 0	77,882	33	0	77,858	33	77,853	159,046	92
EXAVAIL VSAM KSDS	3	0 0	0	210,930	799,098	3,846	27,886	1,692	139,662	2,116
EXBANK VSAM KSDS	3	0 0	0	0	0	157	0	0	151	11
EXEMAIL VSAM KSDS	3	0 0	11,801	828	0	19	809	0	3,915	157
*TOTALS*			89,683	211,791	789,098	81,880	28,728	79,545	302,274	2,376

- Whenever physical I/O has to be done to the VSAM data set, the request is handled via LSR
  - So, it is imperative that the LSR pool to which the data table is assigned be buffered appropriately
    - Preferably, a separate LSR pool **only** for data tables
    - Competition only from other data tables can help maintain the data/index information in buffers longer

# DFH0STAT Output

## Data Tables

### LSR Pools

Pool Number : 3    Time Created : 00:43:12.15918

Maximum key length . . . . . : 117  
 Total number of strings . . . . . : 200  
 Peak concurrently active strings : 4  
 Total requests waited for string : 0  
 Peak requests waited for string. : 0

### Buffer Totals

Data Buffers . . . . . :	800	Index Buffers. . . . . :	700
Successful look asides . . . . . :	930,156	Successful look asides . . . . . :	1,135,894
Buffer reads . . . . . :	<b>111,993</b>	Buffer reads . . . . . :	<b>2,172</b>
User initiated writes. . . . . :	<b>190,781</b>	User initiated writes. . . . . :	204
Non-user initiated writes. . . . . :	0	Non-user initiated writes. . . . . :	0

### Data Buffer Statistics

Size	Buffers	Look Asides	Reads	User Writes	Writes
4096	200	765,785	105,851	33,811	0
8192	100	0	9	142	0
20480	500	164,371	6,133	156,828	0

### Index Buffer Statistics

Size	Buffers	Look Asides	User Reads	Writes	Writes
2048	500	492,882	185	75	0
4096	200	643,012	1,987	129	0

## Look-Aside Hit Ratio

**Data = 89.3%    Index = 99.8%**

**Overall = 94.8%**

### DATA EXCP

**111,993**

**190,781**

**302,774**

### INDEX EXCP

**2,172**

**204**

**2,376**

**Buffer Reads + User Initiated Writes  
 (data and index) = EXCP = Physical  
 I/O**

---

# DFH0STAT

## Closing

- DFH0STAT has some very useful tuning information
  - Not as complete as the EOD statistics
  - Can be very bulky (program/transactions)
- Can be used to tune or problem determination
  - Certain SIT parameters
  - TS/TD
  - Files/LSR
  - Data Tables
  - Virtual Storage (DSALIM and EDSALIM)

---

# DFH0STAT

## Closing

- DFH0STAT desired enhancements
  - Additional information
    - File CISZ for data and index
    - Certain % calculations (e.g., CPU dispatch Ratio)
  - Session information
  - Report selection
    - By Domain
    - By function
      - LSR/File

---

**Thank You Very Much!!!**