



TAKING THE PAIN OUT OF CICS APPLICATION DEBUGGING

OR, HOW TO BE A BETTER PROGRAMMER BY DOING LESS

CICS VIRTUAL USER GROUP
JANUARY 11, 2022

WENDELL LOVEWELL
MACKINNEY SYSTEMS, INC.

THREE QUALITIES OF A GOOD PROGRAMMER

1 - Laziness

You'll go to great lengths to reduce overall energy expenditure

2 - Impatience

The anger you feel when the computer is being lazy

3 - Hubris

Excessive pride that makes you write/maintain programs that others won't say bad things about

~ Larry Wall, original author of the Perl Programming Languages
(Excerpts from the Perl Glossary at <https://perldoc.perl.org/perlglossary>)

HOW TO BE A “BETTER” PROGRAMMER

Your computer can do a gazillion things a second.

How many can you do?

Let the computer do more, so you can do more!

HOW TO GET THE COMPUTER TO DO MORE

Stop doing time-consuming things yourself!

It hurts to do things slowly.

What are you doing that hurts?

“HEE HAW” WISDOM

“Doc, it hurts when I do this!”



The background is a teal-to-blue gradient. In the top-left and top-right corners, there are white line-art graphics resembling circuit traces or neural network connections, with small circles at the end of the lines.

“HEE HAW” WISDOM

“THEN DON’T DO THAT!”

DON'T DO WHAT?

When it comes to diagnosing CICS application problems, what is most painful about the way you resolve them?

- **Reading Dumps?**
- **Analyzing Traces?**
- **Stepping through with Debuggers?**

DOWN IN THE DUMPS?

A transaction abending is probably one of the more common indicators something went wrong.

Depending on system settings, a “dump” can be created.

A dump contains the contents of memory used by the program, and some indication of what instruction the program was trying to execute.

DOWN IN THE DUMPS?

Messages notifying you of an abend can appear:

- On the terminal:

```
DFHAC2206 12:17:08 T560 Transaction KASE failed with abend ASRA. Updates to  
local recoverable resources backed out.
```

DOWN IN THE DUMPS?

Messages notifying you of an abend can appear:

- On the system console:

```
  Display  Filter  View  Print  Options  Search  Help
-----
ISFPCU41 OG  24367.101 S0W2 S0W2 01/10/2022 0W          6,076  COLUMNS 52- 131
COMMAND INPUT ==>                                     SCROLL ==> CSR
0090  +DFHSR0001  T560      An abend (code 0C9/AKEA) has occurred at offset
      X'000001AE' in program ESIPCS33.
0090  +DFHME0116 T560 626
0090  (Module:DFHMEME) CICS symptom string for message DFHSR0001 is
0090  PIDS/5655Y0400 LVLS/730 MS/DFHSR0001 RIDS/DFHSRP PTFS/HCI7300
0090  AB/S00C9 AB/UAKEA RIDS/ESIPCS33 ADRS/000001AE
0090  +DFHDU0205  T560      A SYSTEM DUMP FOR DUMPCODE: SR0001  , WAS
      SUPPRESSED BY THE DUMP TABLE OPTION FOR THIS DUMPCODE
```

DOWN IN THE DUMPS?

Messages notifying you of an abend can appear:

- Into one of the "MSGUSR" Transient Data destinations:

```
MacKinney Systems                                CICS/Log View                                Queue = ALL
Start Date 20220110                                This Applid = T560
Start Time 143600                                View Applid = T560
Command-->
T=TOP B=BOT BB=AUTO-BOT F=_FIND L=LOG S=SHOW

@=CO !=HL #=ME +=PR *=XX                1    1    2    2    3    3    4    4    5    5    6    6    7    7    8    8    9    9    0    0    1
LOGGED    QUEUE 1...5...0...5...0...5...0...5...0...5...0...5...0...5...0...5...0...5...0...5...0...5...0

220110 14:36:08 CDUL DFHDU0203I 01/10/2022 14:36:08 T560 A transaction dump was taken for dumpcode: ASRA, Dumpid: 1/0002, Tranid: K
220110 14:36:08 CDUL                                Program: ESIPCS33.
220110 14:36:08 CSMT DFHAC2236 01/10/2022 14:36:08 T560 Transaction KASE transaction number 00698 abend ASRA in program ESIPCS33 te
220110 14:36:08 CSMT                                local recoverable resources will be backed out.
```

DOWN IN THE DUMPS?

Ok, so there was a problem, and it produced a dump.

What do you have to know?

The background is a gradient of teal and blue. In the corners, there are decorative white lines resembling a circuit board or network diagram, with small circles at the end of the lines.

DOWN IN THE DUMPS?

THAT DEPENDS !

**The first thing you have to know to read a dump is
“Where is it?”**

The background is a teal-to-blue gradient. In the corners, there are decorative white circuit-like lines with small circles at the ends, resembling a PCB layout.

DOWN IN THE DUMPS?

THAT ALSO DEPENDS !

**You also have to know what kind of dump,
(if any)
was produced for the problem.**

TWO TYPES OF CICS DUMPS

Transaction Dump

A “Transaction Dump” contains memory (data and code) used by a single CICS transaction.

Messages will be sent to the CDUL* transient data destination if a transaction dump was produced:

```
MacKinney Systems                               CICS/Log View                               Queue = ALL
Start Date 20220110                             This Applid = T560
Start Time 143600                               View Applid = T560
Command-->
T=TOP B=BOT BB=AUTO-BOT F=FIN L=LOG S=SHOW

@=CO !=HL #=ME +=PR *=XX      1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9  0  0  1
LOGGED  QUEUE 1...5...0...5...0...5...0...5...0...5...0...5...0...5...0...5...0...5...0...5...0
220110 14:36:08 CDUL DFHDU0203I 01/10/2022 14:36:08 T560 A transaction dump was taken for dumpcode: ASRA, Dumpid: 1/0002, Tranid: K
220110 14:36:08 CDUL                               Program: ESIPCS33.
220110 14:36:08 CSMT DFHAC2236 01/10/2022 14:36:08 T560 Transaction KASE transaction number 00698 abend ASRA in program ESIPCS33 te
220110 14:36:08 CSMT                               local recoverable resources will be backed out.
```

* CDUL is often included in the MSGUSR (SYSOUT) DD for the CICS region.

TWO TYPES OF CICS DUMPS

System Dump

A “System Dump” contains most of the memory allocated to an entire CICS region. Because of their large size, system dumps are often suppressed by CICS:

```
  Display  Filter  View  Print  Options  Search  Help
-----
ISFPCU41 OG  24367.101 s0w2 s0w2 01/10/2022 0w          6,076  COLUMNS 52- 131
COMMAND INPUT ==>                                     SCROLL ==> CSR
0090  +DFHSR0001  T560      An abend (code 0C9/AKEA) has occurred at offset
      X'000001AE' in program ESIPCS33.
0090  +DFHME0116  T560 626
0090  (Module:DFHMEME) CICS symptom string for message DFHSR0001 is
0090  PIDS/5655Y0400 LVLS/730 MS/DFHSR0001 RIDS/DFHSRP PTFS/HCI7300
0090  AB/S00C9 AB/UAKEA RIDS/ESIPCS33 ADRS/000001AE
0090  +DFHDU0205  T560      A SYSTEM DUMP FOR DUMPCODE: SR0001 , WAS
      SUPPRESSED BY THE DUMP TABLE OPTION FOR THIS DUMPCODE
```

WHAT DO I HAVE TO KNOW TO USE TRANSACTION DUMPS?

- There are 2 dump datasets, “DFHDMPA” and “DFHDMPB”
- Both are defined in the JCL for the CICS region
- Only one is active at a time
- Use the CEMT SET DUMP SWITCH command to control:

```
SET DUMP SWI █  
STATUS: RESULTS - OVERTYPE TO MODIFY  
Dum Cur(B) Ope Aut
```

At a minimum, you have to know the transaction ID and which Dump Data Set (“A” or “B”) was used.

WHAT DO I HAVE TO KNOW TO USE TRANSACTION DUMPS?

You also have to have JCL to “print” the dump:

```
//DFHDUMPS JOB CLASS=A
//PRTDUMP EXEC PGM=DFHDU730, PARM=SINGLE, REGION=0M
//STEPLIB DD DISP=SHR, DSN=CICSTS56.CICS.SDFHLOAD
//DFHDMPDS DD DISP=SHR, DSN=CICSTS56.T560.DFHMPA ←
//SYSPRINT DD SYSOUT=*
//DFHTINDX DD SYSOUT=*
//DFHPRINT DD SYSOUT=*
//SYSIN DD *
SELECT TYPE=OR
TRANID=KASE ←
END
//*
```

Notes:

- The suffix for DFHDUxxx is the 3-digit CICS “release” level (TS5.5 = 720, TS5.6 = 730, etc)
- DFHDMPDS must specify the correct (“A” or “B”) dataset
- You can specify inclusion criteria, such as transaction ID to be printed.
- Refer to <https://www.ibm.com/docs/en/cics-ts/5.6?topic=program-format-sysin-control-statements>

WHAT DO I HAVE TO KNOW TO USE TRANSACTION DUMPS?

The bad news is: Printing a dump is the easy part.

What do you have to know to use (aka “read”) a dump?

WHAT DO I HAVE TO KNOW TO USE TRANSACTION DUMPS?

The bad news is: Printing a dump is the easy part.

What do you have to know to use (aka “read”) a dump?

A LOT !!!

WHAT DO I HAVE TO KNOW TO USE TRANSACTION DUMPS?

```
L:\STD\DFHDUMPS.ASMLIST      Line=7      Col=1      Size=3064 Alt=0,0;2
====> |
T560      --- CICS TRANSACTION DUMP ---  CODE=ASRA  TRAN=KASE  ID=1/0002  DATE=22/01/10  TIME=14:36:08  PAGE  1
SYMPTOMS= AB/UASRA PIDS/5655Y0400 FLDS/DFHABAB RIDS/ESIPCS33
CICS LEVEL = 0730
PSW & REGISTERS AT TIME OF INTERRUPT
PSW:      07950000  A137300E  00020009  02A8B400
PSW16:    07950000  80000000  00000000  2137300E
TRANSLATION EXCEPTION ADDRESS: 00000000_02A8B400
BREAKING EVENT ADDRESS:      00000000_21372FFC
ADDRESSING MODE: 31
REGS 0-3  00000000_21100A28  00000000_21100908  00000000_00041800  00000000_A1372E88
REGS 4-7  00000000_00000000  00000000_00000000  00000000_2044BE4C  00000000_20E2F320
REGS 8-B  00000000_1FC94DD8  00000000_211008A0  00000000_21100008  00000000_21100100
REGS C-F  00000000_008B3000  00000000_21100048  00000000_A1372FDA  00000000_00000000
EXECUTION KEY 9
The transaction was in Subspace mode
REGISTERS AT LAST EXEC COMMAND
REGS 0-7  21100A28  21100908  00041800  A1372E88  00000000  20E2F13C  2044BE4C  20E2F320
A137300E - A1372E88 = 390 (hex 186)
```

What does all this mean?

How long does it take to learn how to use these?

Once I've learned, How long does it take to determine the problem?

WHAT DO I HAVE TO KNOW TO USE TRANSACTION DUMPS?

```
L:\STD\DFHDUMPS.ASMLIST      Line=2777 Col=7  Size=3064 Alt=0,0;2
====>
PROGRAM STORAGE              ADDRESS 21372E60 TO 213730EF  LENGTH 00000290
00000000  C4C6C8E8 C1F4F1F0 58F0021C 58F0F0D0 58F0F014 58F0F00C 58FF000C 07FF5CC6 *DFHYA410.0...00..00..00.....*F*
00000020  C9D3D3C9 D55C0000 90ECD00C 183F4510 300E018F 000058F0 320805EF 50D01004 *ILLIN*.....0.....*
00000040  18F1BF1F D0184780 3028D207 F05C1000 189F58B0 905C5840 906047F0 3076C5E2 *.1.....K.0*.....*..-.0..ES*
00000060  C9D7C3E2 F3F3406C E5D940C3 88819587 858440F0 F461F1F9 61F2F0F1 F240C1A2 *IPCS33 .VR Changed 04/19/2012 As*
00000080  A2859482 93858440 969540F0 F161F1F0 61F2F240 F1F44BF2 F7404040 40404110 *sembled on 01/10/22 14.27 ..*
000000A0  906841E0 322841F0 320C90EF 10009680 100458F0 321050D0 905441D0 900005EF *.....0.....o.....0.....*
000000C0  58D09054 D503B008 32144780 310ED503 B0083218 4780310E 92E4918E 41109068 *...N.....N.....kuj.....*
000000E0  41E031F8 41F0B010 41009184 90E01000 96801008 58F03210 50D09054 41D09000 *..8.0....jd....o....0.....*
00000100  05EF58D0 9054D503 9184321C 478030FE D5039184 32204780 30FA92E3 918E47F0 *.....N.jd.....N.jd.....kTj..0*
00000120  30FE92D4 918E95D4 40004780 312295E3 40004780 3118D203 9188321C 47F0312C *..kmj.nM .....nT .....K.jh...0..*
00000140  D2039188 322447F0 312CD203 91883220 47F0312C 41109068 41E03200 41F0B010 *K.jh...0..K.jh...0.....0..*
00000160  41009188 90E01000 96801008 58F03210 50D09054 41D09000 05EF58D0 905447F0 *..jh....o....0.....0.....0*
00000180  315AD501 B018323E 4780316E D2004000 918E47F0 318AD503 B0083214 47803182 *..N.....>K. .j..0..N.....b*
000001A0  D503B008 32184770 318A1B55 1D0547F0 318A4110 906841E0 324050E0 10009680 *N.....0.....o.....*
000001C0  100058F0 321050D0 905441D0 900005EF 58D09054 D396A685 99608381 A2854083 *...0.....Lower-case c*
000001E0  88819981 83A38599 A240A689 93934082 8540A399 8195A293 81A38584 40A39640 *haracters will be translated to *
00000200  A4979785 99408381 A2854B00 58D09004 1B1158F0 320805EF 98ECD00C 07FE0000 *upper case.....0....q.....*
```

What does all this mean?

Offset x'1AE' -> instruction x'1D05'

WHAT DO I HAVE TO KNOW TO USE TRANSACTION DUMPS?

```
C:\Temp\ESIPCS33.LISTING      Line=480  Col=7  Size=624  Alt=0,0;0*
====>
00015A D501 B018 323E 00018 0023E 327      CLC  EIBCALEN,=H'0'      ANY COMM AREA PASSED?
000160 4780 316E                0016E 328      BE   TERMMSG           NO, THE DISPLAY MSG.
000164 D200 4000 918E 00000 0018E 329      MVC  UCTRNC D,PREVCASE   Yes, change to the new setting
00016A 47F0 318A                0018A 330      B   OUTAHERE
                0016E 331 TERMMSG EQU *
00016E D503 B008 3214 00008 00214 332      CLC  EIBTRNID,=C'KASE'
000174 4780 3182                00182 333      BE   SENDUPPR
000178 D503 B008 3218 00008 00218 334      CLC  EIBTRNID,=C'kase'
00017E 4770 318A                0018A 335      BNE  OUTAHERE
                00182 336 SENDUPPR EQU *
                337 *      EXEC  CICS SEND TEXT FROM(UPPRMSG) LENGTH(55) NOHANDLE ERASE
                338 * 2007/10/27: Don't make user clear screen to continue
                339 *!      EXEC  CICS SEND CONTROL ERASE FREEKB
                340 * 2022/01/11: Force an abend for demo purposes
000182 1B55                341      SR   5,5
000184 1D05                342      DR   0,5
000186 47F0 318A                0018A 343      B   OUTAHERE
                344 *
```

X'1D05' = Divide reg 0 by reg 5

Reg 5 = 0

Divide by zero exception

WHAT DO I HAVE TO KNOW TO USE SYSTEM DUMPS?

There are few things you should know about system dumps:

- 1) If you really need one, you'll probably have to enable CICS for it.

```
+DFHDU0205  T560      A SYSTEM DUMP FOR DUMPCODE: SR0001 , WAS  
SUPPRESSED BY THE DUMP TABLE OPTION FOR THIS DUMPCODE
```

Our's today is an SR0001 dump (this time—there are numerous types).

Use CEMT SET SYDUMPCODE to set the MAX count > Cur count

```
SET SYDUMPCODE(SR0001) SYSDUMP MAX(3) =  
STATUS: RESULTS - OVERTYPE TO MODIFY  
Syd(SR0001 ) Sys Loc      Max( 003 ) Cur(0002)
```

(The “Max” value is the total number of times the dump will be written.

You must set it to some number higher than what has already occurred—shown by the “Cur” value.)

WHAT DO I HAVE TO KNOW TO USE SYSTEM DUMPS?

Recreate the problem. The console should indicate a system dump was taken:

```
+DFHSR0001 T560 An abend (code 0C3/AKEA) has occurred at offset
X'000001D0' in program ESIPCS33.
+DFHME0116 T560 110
(Module:DFHMEME) CICS symptom string for message DFHSR0001 is
PIDS/5655Y0400 LVLS/730 MS/DFHSR0001 RIDS/DFHSRP PTFS/HCI7300
AB/S00C3 AB/UAKEA RIDS/ESIPCS33 ADRS/000001D0
+DFHDU0201 T560 ABOUT TO TAKE SDUMP. DUMPCODE: SR0001 , DUMPID:
1/0022
IEA045I AN SVC DUMP HAS STARTED AT TIME=14.49.27 DATE=01/05/2022 112
FOR ASID (0054)
QUIESCE = YES
IEA794I SVC DUMP HAS CAPTURED: 113
DUMPID=001 REQUESTED BY JOB (T560 )
DUMP TITLE=CICS DUMP: SYSTEM=T560 CODE=SR0001 ID=1/0022
+DFHDU0202 T560 SDUMPX COMPLETE. SDUMPX RETURN CODE X'00'
IEF196I IGD100I 0AAA ALLOCATED TO DDNAME SYS00002 DATACLAS (DEFAULT)
IEF196I IEF285I SYS1.SVCDUMP.Z2.D220105.T204927.S00001 CATALOGED
IEF196I IEF285I VOL SER NOS= WORKZ1.
IEA611I COMPLETE DUMP ON SYS1.SVCDUMP.Z2.D220105.T204927.S00001 118
DUMPID=001 REQUESTED BY JOB (T560 )
FOR ASID (0054)
INCIDENT TOKEN: ADCDPL S0W2 01/05/2022 20:49:27
IEF196I IEF237I 0A83 ALLOCATED TO IPCSDDIR
```

The memory of the entire CICS region was written to a sequential dataset named
SYS1.SVCDUMP.Z2.D220105.T204927.S00001

WHAT DO I HAVE TO KNOW TO USE SYSTEM DUMPS?

2) The second thing to know about system dumps is that they are BIG!

```
ISRUAIES                               Data Set Information
Command ==> _

Data Set Name . . . . . : SYS1.SVCDUMP.Z2.D220105.T204927.S00001

General Data                               Current Allocation
Management class . . . . : **None**       Allocated tracks . . : 24,747
Storage class . . . . . : **None**       Allocated extents . . : 1
Volume serial . . . . . : WORKZ1
Device type . . . . . : 3390
Data class . . . . . : **None**
Organization . . . . . : PS
Record format . . . . . : FBS
Record length . . . . . : 4160
Block size . . . . . : 24960
1st extent tracks . . . . : 24747
Secondary tracks . . . . . : 29768
Data set name type . . . . :
Data set encryption : NO

Current Utilization
Used tracks . . . . . : 24,747
Used extents . . . . . : 1

Dates
Creation date . . . . . : 2022/01/05
Referenced date . . . . : 2022/01/05
Expiration date . . . . . : ***None***
```

WHAT DO I HAVE TO KNOW TO USE SYSTEM DUMPS?

The main thing to know about system dumps is...

WHAT DO I HAVE TO KNOW TO USE SYSTEM DUMPS?

The main thing to know about system dumps is...

Don't do that!

WHAT DO I HAVE TO KNOW TO USE SYSTEM DUMPS?

The main thing to know about system dumps is...

Don't do that!

IBM or a vendor (like MacKinney Systems) might ask you for one. (Be sure they ask for a “system” or “svc” dump—a transaction dump may be all they need.)

If they do, you will need to use TRSMAIN or another compression product on the file before it can be processed by the receiving end. Your vendor should give you the necessary instructions.

DUMPS RECAP

What are the advantages of using dumps?

- “Free”--every z/OS system can produce a dump
With training, IPCS can be used to speed the process.
- No setup prior to the problem is needed
(Your system needs to have space available for the dump.)
- Sometimes they are the only way to resolve an issue

DUMPS RECAP

What are the disadvantages to using dumps?

- Reading a dump is usually very labor-intensive.

In other words, they are time-consuming.

(There are vendor products, and IBM's IPCS, which can be used to simplify using a dump.)

- A great deal of technical knowledge is required to use them.

- They are “point in time” only.

It can be difficult to determine program flow prior to theabend.
However, the trace table entries included in the dump can help.

WHAT DO I HAVE TO KNOW TO USE CICS TRACE?

CICS has an integrated tracing facility that can record every interaction of your program with CICS.

- Trace entries are recorded in two locations: Internal and External
- The internal trace is usually active, but tends to “wrap” quickly.
- There are two external (“AUX”) trace files: “DFHAUXT” and “DFHBUXT”

WHAT DO I HAVE TO KNOW TO USE CICS TRACE?

The CETR transaction is used to control the Trace Facility

```
CETR                CICS Trace Control Facility                T560 T560
Type in your choices.

Item                Choice                Possible choices
Internal Trace Status  ==> STARTED                STArtd, STOpped
Internal Trace Table Size ==> 4096 K                1024K - 1048576K

Auxiliary Trace Status ==> STOPPED                STArtd, STOpped, PAused
Auxiliary Trace Dataset ==> A                        A, B
Auxiliary Switch Status ==> NEXT                        NO, NExt, All

GTF Trace Status      ==> STOPPED                STArtd, STOpped

Master System Trace Flag ==> ON                        ON, Off
Master User Trace Flag  ==> ON                        ON, Off

When finished, press ENTER.

PF1=Help  3=Quit  4=Components  5=Ter/Trn  9=Error List
```

WHAT DO I HAVE TO KNOW TO USE CICS TRACE?

You usually have to start it. Note which Dataset is used:

```
CETR                CICS Trace Control Facility                T560 T560

Type in your choices.

Item                Choice                Possible choices
Internal Trace Status  ==> STARTED                STArtd, STOpped
Internal Trace Table Size ==> 4096 K                1024K - 1048576K

Auxiliary Trace Status ==> started                STArtd, STOpped, Paused
Auxiliary Trace Dataset ==> A                    A, B
Auxiliary Switch Status ==> NEXT                NO, NExt, All

GTF Trace Status      ==> STOPPED                STArtd, STOpped

Master System Trace Flag ==> ON                    ON, Off
Master User Trace Flag  ==> ON                    ON, Off

When finished, press ENTER.

PF1=Help  3=Quit  4=Components  5=Ter/Trn  9=Error List
```

WHAT DO I HAVE TO KNOW TO USE CICS TRACE?

To avoid recording activity for everything in CICS, you can be selective:

```
CETR                               Transaction and Terminal Trace                               T560 T560
Type in your choices.
Item                                Choice                                Possible choices
Transaction ID                      ==> kase                             Any valid 4 character ID
Transaction Status                   ==> -                                Standard, Special, Suppressed
Terminal ID                          ==>                                 Any valid Terminal ID
Netname                              ==>                                 Any valid Netname
Terminal Status                      ==>                                 Standard, Special
Terminal VTAM Exit Trace             ==>                                 ON, OFF
Terminal ZCP Trace                   ==>                                 ON, OFF
VTAM Exit override                   ==> NONE                             All, System, None

When finished, press ENTER.

PF1=Help    3=Quit    6=Cancel Exits    9=Error List
```

WHAT DO I HAVE TO KNOW TO USE CICS TRACE?

Run your test. Then stop the trace ASAP:

```
CETR                CICS Trace Control Facility                T560 T560
Type in your choices.
Item                Choice                Possible choices
Internal Trace Status  ==> STARTED                STArted, STOpped
Internal Trace Table Size ==> 4096 K                1024K - 1048576K
Auxiliary Trace Status ==> stopped                STArted, STOpped, Paused
Auxiliary Trace Dataset ==> A                        A, B
Auxiliary Switch Status ==> NEXT                NO, NExt, All
GTF Trace Status      ==> STOPPED                STArted, STOpped
Master System Trace Flag ==> ON                    ON, OFF
Master User Trace Flag ==> ON                    ON, OFF
When finished, press ENTER.

PF1=Help  3=Quit  4=Components  5=Ter/Trn  9=Error List
```


WHAT DO I HAVE TO KNOW TO USE CICS TRACE?

Now what?

Trace dataset DFHAUXT now contains records for your trace.

You must “print” them using program DFHTU_{nnn}
where _{nnn} = system release (e.g. TS5.6 = ‘730’)

```
//PRTAUXT EXEC PGM=DFHTU730,REGION=0M
//STEPLIB DD DISP=SHR,DSN=CICSTS56.CICS.SDFHLOAD
//          DD DISP=SHR,DSN=CICSTS56.CICS.SDFHLINK
//DFHAUXT DD DISP=SHR,DSN=CICSTS56.T560.DFHAUXT ←
//DFHAXPRT DD SYSOUT=L
//DFHAXPRM DD *
FULL,
PAGESIZE=9999,
TRANID=KASE ←
//* FULL          FULL|SHORT|ABBREV.  ABBREV IS 1 ENTRY PER LINE
//* TIMERG=(093000-11300),
//* TERMID=T03A,
//* TASKID=1208,
//* TYPETR=(AP006E-0120)
//* ALL          PRINT ALL TRACE ENTRIES IN TRACE DATA SET!
//* EXCEPTION   PRINT ONLY EXCEPTIONS
//
```

<https://www.ibm.com/docs/en/cics-ts/5.6?topic=programdfhtu730-trace-selection-parameters-trace-utility-program>

WHAT DO I HAVE TO KNOW TO USE CICS TRACE?

Using an AUXTRACE listing is like looking for a needle in a haystack...

```
00011 CICS - AUXILIARY TRACE FROM 01/06/22 - APPLID T560 PAGE 00002
00012
00013
00014 XM 1102 XMAT EXIT - FUNCTION(ATTACH) RESPONSE(OK) TRANSACTION_TOKEN(21036700 , 0000119C) TRANNUM(0000119C)
00015
00016 TASK-TCP KE_NUM-0020 TCB-C/QR /008FC718 RET-A04F9A78 TIME-11:18:17.7242041250 INTERVAL-00.0000053281 =0000180
00017 1-0000 01100000 000000A0 00000000 00000000 B8A58540 02A00000 01000100 D2C1E2C5 *.....ve .....KASE*
00018 0020 00000000 1FC99800 45000000 10001000 00000000 1F942226 00000200 2002C580 *.....Iq.....m.....E.*
00019 0040 21036700 0000119C 0000119C 1F9424AE 00444001 00000000 00000000 20F97000 *.....m.....9..*
00020 0060 00000000 00017002 00000000 21035570 02A00000 00000000 00000000 9FA59CB6 *.....v..*
00021 0080 00000000 1F95A968 00000000 9FA59CE4 1FAF9B40 1FB09C80 1FBADC00 1FBDD8E4 *.....nz.....v.U.....QU*
00022 00A0 1FBA8000 1FBBA020 00000028 00000000 01020002 0E02E528 00000000 1F94228A *.....V.....m..*
00023 00C0 00700000 00000028 00000000 00000000 00000000 9FA59C7E 01000000 1F94563E *.....v.=.....m..*
00024 00E0 00000000 0C0224AE 20F9C280 00000098 00000100 20F9C998 00000000 1F9422A8 *.....9B.....q.....9Iq.....m.y*
00025 0100 00700000 00000028 00000000 00000000 *.....*
00026
00027 AP EA00 TMP ENTRY FUNCTION(LOCATE) TABLE(PFT ) KEY(DFHICST)
00028
00029 TASK-00119 KE_NUM-0046 TCB-C/QR /008FC718 RET-A0600F34 TIME-11:18:17.7250680312 INTERVAL-00.0000164062 =0000199
00030 1-0000 01000400 20E80CA8 00000000 00000000 00000000 *.....Y.y.....*
00031 2-0000 C4C6C8C3 C9C3E2E3 *DFHICST *
00032
00033 AP EA01 TMP EXIT FUNCTION(LOCATE) TABLE(PFT ) KEY(DFHICST) ENTRY_ADDRESS(2145B500) RESPONSE(NORMAL)
00034
00035 TASK-00119 KE_NUM-0046 TCB-C/QR /008FC718 RET-A0600F34 TIME-11:18:17.7250804531 INTERVAL-00.0000124218 =0000200
00036 1-0000 01000400 20E80CA8 2145B500 00000000 00000000 *.....Y.y.....*
00037 2-0000 C4C6C8C3 C9C3E2E3 *DFHICST *
```

A really big haystack!

Almost everything CICS does under the covers is exposed.
Each domain call is shown with matching ENTRY and EXIT displays.

WHAT DO I HAVE TO KNOW TO USE CICS TRACE?

With an editor (or IPCS), you can find *EXC* entries for the exception...

```
01012 AP 1942 APLI *EXC* - Program-Check FUNCTION(START_PROGRAM) PROGRAM(ESIPCS33) CEDF_STATUS(CEDF) EXECUTION_SET(FULLAPI)
01013 ENVIRONMENT_TYPE(EXEC) SYNCONRETURN(NO) LANGUAGE_BLOCK(2156DC10) COMMAREA(00000000 , 00000000) LINK_LEVEL(1)
01014 SYSEIB_REQUEST(NO)
01015
01016 TASK-00119 KE_NUM-0046 TCB-C/QR /008FC718 RET-9FF62F66 TIME-11:18:17.7620424218 INTERVAL-00.0240092968* =0000321
01017 1-0000 01B80000 000000DA 00000000 00000000 B81B4750 00000000 02000100 C5E2C9D7 *.....&.....ESIP*
01018 0020 C3E2F3F3 2131B000 A131B000 000002B8 02010101 01020202 2156DC10 2156DC10 *CS33.....*
01019 0040 000002B8 1F95CC88 00000050 00256000 00000000 00000000 00000000 00010102 *.....n.h...&...-.....*
01020 ...
01031 01A0 00000000 00000000 B5800000 00000000 01041800 20E81860 *.....Y.- *
01032 2-0000 F0C3F361 C1D2C5C1 018700C3 00000000 C4C6C8C1 D7D3C9F1 00000000 20450900 *0C3/AKEA.g.C....DFHAPLI1.....*
01033 0020 00000000 1FCA3800 00000000 20E7E000 51CBB680 00000003 00000003 FFFFFFFF *.....X\.....*
01034 ...
01071 04C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
01072 04E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
01073
01074 AP 0790 SRP *EXC* - PROGRAM_CHECK
01075
01076 TASK-00119 KE_NUM-0046 TCB-C/QR /008FC718 RET-9FC953EC TIME-11:18:17.7620580625 INTERVAL-00.0000156406 =0000322
01077 1-0000 F0C3F361 C1D2C5C1 018700C3 00000000 C4C6C8C1 D7D3C9F1 00000000 20450900 *0C3/AKEA.g.C....DFHAPLI1.....*
01078 0020 00000000 1FCA3800 00000000 20E7E000 51CBB680 00000003 00000003 FFFFFFFF *.....X\.....*
01079 ...|
01117 04E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
01118
01119 DS 0010 DSBR ENTRY - FUNCTION(INQUIRE_TASK)
01120
```

But without help or a lot of experience, it's tough.

WHAT DO I HAVE TO KNOW TO USE CICS TRACE?

And even when you get there, it may not be very helpful...

```
01012 AP 1942 APLI *EXC* - Program-Check FUNCTION(START_PROGRAM) PROGRAM(ESIPCS33) CEDF_STATUS(CEDF) EXECUTION_SET(FULLAPI)
01013 ENVIRONMENT_TYPE(EXEC) SYNCONRETURN(NO) LANGUAGE_BLOCK(2156DC10) COMMAREA(00000000 , 00000000) LINK_LEVEL(1)
01014 SYSEIB_REQUEST(NO)
01015
01016 TASK-00119 KE_NUM-0046 TCB-C/QR /008FC718 RET-9FF62F66 TIME-11:18:17.7620424218 INTERVAL-00.0240092968* =0000321
01017 1-0000 01B80000 000000DA 00000000 00000000 B81B4750 00000000 02000100 C5E2C9D7 *.....&.....ESIP*
01018 0020 C3E2F3F3 2131B000 A131B000 000002B8 02010101 01020202 2156DC10 2156DC10 *CS33.....*
01019 0040 000002B8 1F95CC88 00000050 00256000 00000000 00000000 00000000 00010102 *.....n.h...&..-.....*
01020 0060 00020201 03E81590 00020000 1FBE04B8 01000000 1FAE6800 00000000 20001090 *.....Y.....*
01021 0080 00000000 20E81090 00000000 20E81090 00000000 00B4D8C0 C5E2C9D3 C9C24040 *.....Y.....Y.....Q{ESILIB *
01022 00A0 00000000 1FAE4000 80000000 00000000 00000000 20E81CB0 00000000 1F95CC88 *.....Y.....Y.....n.h*
01023 00C0 00000000 20E81590 00000000 20E818D0 00000050 4225D480 00000050 4225D480 *.....Y.....Y.}.&..M...&..M.*
01024 00E0 02020000 1FADF800 80000000 00000000 00000000 20E81C40 00000000 1F955950 *.....8.....Y.....n.&*
01025 0100 00000000 20E812C0 20E81C8C 20E81E64 1FBBF000 51CB9380 1FBC0000 00000000 *.....Y.{.Y...Y...0...l...*
01026 0120 00000000 20E81F60 20E81F60 00007000 00000000 20E81CE3 00F3C1C4 20E81798 *.....Y.-.Y.-.....Y.T.3AD.Y.q*
01027 0140 54000000 00000000 00000000 A0C51E4A 00000000 A0C5215A 00000000 00001000 *.....E.J.....E.!.....*
01028 0160 0000005A 00000000 1FBA4000 1FB09C80 1FBE2D80 51CAD1B8 00000000 1FBA8000 *.....!.....J.....*
01029 0180 00000000 00000000 00000000 00000000 F10005A0 C6F4E2C1 00055000 00000000 *.....1...F4SA..&....*
01030 01A0 00000000 00000000 B5800000 00000000 01041800 20E81860 *.....Y.-.....*
01031 2-0000 F0C3F361 C1D2C5C1 018700C3 00000000 C4C6C8C1 D7D3C9F1 00000000 20450900 *OC3/AKEA.g.C...DFHAPLI1.....*
01032 0020 00000000 1FCA3800 00000000 20E7E000 51CBB680 00000003 00000003 FFFFFFFF *.....X\.....*
01072 AP 0790 SRP *EXC* - PROGRAM_CHECK
01116 DS 0010 DSBR ENTRY - FUNCTION(INQUIRE_TASK)
01125 DS 0011 DSBR EXIT - FUNCTION(INQUIRE_TASK) RESPONSE(OK) ESSENTIAL_TCB(ESSENTIAL_YES)
01134 PG 0500 PGIS ENTRY - FUNCTION(INQUIRE_CURRENT_PROGRAM)
01152 PG 0501 PGIS EXIT - FUNCTION(INQUIRE_CURRENT_PROGRAM) RESPONSE(OK) CURRENT_PROGRAM_LENGTH(2B8) CURRENT_LOAD_POINT(2131B000)
01171 AP 0781 SRP *EXC* - ABEND_ASRA PROGRAM(ESIPCS33) OFFSET(000001D0) EXEC_KEY(USER)
01217 MN 0D01 MNAC ENTRY - FUNCTION(INQUIRE_CURRENT_ACD) APPLICATION_NAME(20E833A4 , 00000000 , 00000040)
```

This 65-statement program (5 with EXEC CICS statements)
generated around 1600 trace entries with nearly 20,000 print lines!

WHAT DO I HAVE TO KNOW TO USE CICS TRACE?

There are products, including IPCS, that help you decipher the trace data. But without a lot of knowledge (i.e. “experience”), you may find trace data, like system dumps, something you only send to IBM or a vendor.

One thing also to note—DFHDUnnn (the dump “print” program) includes an abbreviated trace table following the dump of memory.

CICS TRACE RECAP

What are the advantages of using AUXTRACE?

- “Free”—built into every version of CICS
With training, IPCS can be used to speed the process.
- No setup prior to the problem is needed
(Depending on the application, you may need large trace datasets)
- Sometimes they are the only way to resolve an issue

CICS TRACE RECAP

What are the disadvantages to using AUXTRACE?

- A great deal of technical knowledge is required to use them.
There are products that make it easier.
- There is a lot of data generated
I already mentioned the 5 “EXEC CICS” statement program generated 20,000 lines of trace data.
- It’s likely to take a long time to find a problem.
“Can’t see the forest for all the trees!”

WHAT ABOUT DEBUGGERS?

A third type of tool in application diagnosis is the “Debugger”.

A debugger is an add-on product that allows you to step through your program, generally one statement at a time. You can display and change the program’s data, and alter the sequence statements that get executed.

WHAT DO I HAVE TO KNOW TO USE DEBUGGERS?

- Debuggers work by matching a program's executable code with the matching compile or assembly listing.
- When the machine code for a particular statement is about to be executed, the debugger will position your display to that statement in the compile/assembly listing.
- This requires that the debugger at some time have access to either the ADATA generated by the compiler/assembler, or the listing.
- There is often a step added to the compiler JCL to make it ready for debugging.

WHAT DO I HAVE TO KNOW TO USE DEBUGGERS?

- Debuggers are real-time: You cannot use a debugger to diagnose a problem that occurred previously.
- The debugger must be told to start debugging a program. Each debugger has its own way to do this.
- The user interface can be a 3270 session or web browser.

WHAT DO I HAVE TO KNOW TO USE DEBUGGERS?

Here's an example debug window:

```
=5.1 TRACK - stop Display WRL1 T03R 01/06/22 14.20.26
Command-----> _
Offset-----> Password----->

Stop Program--> ESIPCS33 Current Module-> ESIPCS33 At Terminal---> T03R
Phase Offset--> 000028 Module offset--> 000000 Statement No.-> 000179
Stop Reason--> HALT REQUEST - HALT ID = 1
  stmt  ....+...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8....+...9....+...10....+...11..
000176          +*          PROLOG CODE FOR EXEC INTERFACE          *
000177          +*****
000178 000000          00000 00278 +ESIPCS33 CSECT          @BBAC81A 01-DF
H00179 000000 90EC D00C          0000C + STM 14,12,12(13)          SAVE CALLER'S REGISTERS @BBAC81A 01-DF
-----Column-> 1-----Type-> Value-----
GPR E-C          E 0008B914 F 2131B028 0 1FBA4000 1 2260003C 2 00041800 3 20457680 4 20E79CC0 5 20E7A13C          >
Stg at 22600054 Le  ....U...R...-..... ..U...<..W..H.Q.X-...VM....

000180          +*****
000181          +*          ESTABLISH CODE ADDRESSIBILITY          *
000182          +*****
000183          +          USING #-4,3          @BBAC81A 01-DF
000184 000004 183F          +          LR 3,15          @BBAC81A 01-DF
000185          +*****
000186          +*          OBTAIN DYNAMIC STORAGE          *
000187          +*****
000188 000006 4510 300E          0000E +          BAL 1,#+8          @L2A 01-DF
000189 00000A 018F          +          DC AL2(DFHEIEND-DFHEISTG) LENGTH OF STORAGE          @L2A 01-DF
000190 00000C 0000          +          DC H'0' IDENTIFIES CICS 1.7 LEVEL          @L2A 01-DF
000191 00000E 58F0 3228          00228 +          L 15,=V(DFHEAT0)          @F8E1S @L2A 01-DF
000192 000012 05EF          +          BALR 14,15          OBTAIN STORAGE          @F8E1S @L2A 01-DF
000193 000014 50D0 1004          00004 +          ST 13,DFHEISA-DFHEISTG+4(,1) CHAIN TO CALLER'S          @F8E1S @L2A 01-DF
000194 000018 18F1          +          LR 15,1          ADDRESS OF STORAGE          @F8E1S @L2A 01-DF
000195 00001A BF1F D018          00018 +          ICM 1,B'1111',24(13) RESTORE INPUT REGISTER 1 @F8E1S @L2A 01-DF
000196 00001E 4780 3028          00028 +          BZ #+10          IF ARGUMENT LIST EXISTS          @L2A 01-DF
```

WHAT DO I HAVE TO KNOW TO USE DEBUGGERS?

You can step through the code one line at a time, or continue to selected “halt points” to avoid stepping through each line.

```
=5.1 TRACK - Stop Display WRL1 T03R 01/06/22 14.55.42
Command-----> _
Offset-----> Password----->

Stop Program--> ESIPCS33 Current Module-> ESIPCS33 At Terminal--> T03R
Phase offset--> 000136 Module offset--> 00010E Statement No.-> 000284
Stop Reason--> HALT REQUEST - HALT ID = 2
 Stmt  ....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9....+...10....+...11..
000281                                     # *****
000282                0010E      SETUPPER EQU #
000283                                     #      MVC   NEWCASE,DFHVALUE(UCTRAN)
H00284 00010E D203 9188 321C 00188 0021C      MVC   NEWCASE,=F'450'
-----Column-> 1-----Type-> Hex-----
Stg at 21100A28 Len 04
                00000000
Stg at 21322144 Len 04
                000001C2
-----Column-> 1-----Type-> Hex-----
NEWCASE(9)      00000000
```

Variables can be displayed and changed, either as they change, on request, or all the time.

WHAT DO I HAVE TO KNOW TO USE DEBUGGERS?

If the program is abending, you can just continue until the abend occurs.

```
=5.1 TRACK - stop display WRL1 T03R 01/06/22 15.01.13
Command-----> -
Offset-----> Password---->

Stop Program--> ESIPCS33 Current Module-> ESIPCS33 At Terminal---> T03R
Phase Offset--> 0001AC Module Offset--> 000184 Statement No.-> 000343
Stop Reason--> FIXED-POINT DIVIDE
  STMT  ....+...1....+...2....+...3....+...4....+...5....+...6....+...7....+...8....+...9....+...10....+...11..
000340                                     # 2022/01/11: Force an abend for demo purposes
000341                                     # EX 0,# d2201
000342 000182 1B55                          SR 5,5
000343 000184 1D05                          DR 0,5
-----Column-> 1-----Type-> Hex-----
GPR 0,1 0 21100A28 1 21100908
GPR 5 5 00000000
-----Column-> 1-----Type-> Hex-----
NEWCASE(9) 000001C2
000344 000186 47F0 318A 0018A B OUTAHERE
```

You can also change the order of the program statements.

DEBUGGER RECAP

What are the advantages to using debuggers?

- You can observe your program executing.
Without a debugger, you either must desk check manually, or test the best you can and hope you didn't miss anything.
- You can change data on-the-fly to test exceptions.
Some exceptions are difficult to produce.
- You can verify the results of every statement.
Or, you can skip to the interesting parts.
- It's fairly easy
If you can code, you will understand the debugger output.

DEBUGGER RECAP

What are the disadvantages to using debuggers?

- It can take a very long time to observe every line of code! Functions like “continue” and “halt” can reduce this, but debugging sessions tend to take a long time.
- Sometimes, observing the program changes the results. Particularly, timing-sensitive issues.
- You generally have to be “live and in person”. If something fails when you’re not there, a debugger won’t show you what happened and you may have to read a dump anyway.

RECAP

So far, we've learned that:

- Reading a dump is difficult and slow
 - but all the point-in-time info is there.
- Reading a CICS trace is really difficult, and is really slow
 - but everything that happened is there.
- Stepping thru a program with a debugger is easy
 - but can be really slow.

RECAP

Is there a better way?

WHAT WE REALLY WANT

What if you could combine:

- The ability to capture everything that happened, like a trace.
- The ability to display contents of any or all variables, like a dump
But be able to see not only all the variables every “step”, but let the tool compare & point out the differences each step made!
- The ability to display every step, like a debugger, but

NOT HAVE TO WAIT FOR IT!

WHAT WE REALLY WANT

We would like to announce a new paradigm for program development!

Introducing:

CICS Task Recorder!

CICS TASK RECORDER

With CICS Task Recorder you can:

- Examine the execution flow of a CICS task
 - Like a debugger, but you can go straight to points of interest
- Display the contents of selected or all Working Storage variables
 - Like a dump, except you can examine the variables at each step
- Compare the results of 2 runs of the same transaction
 - Showing both any differences in steps, and the value of variables
- Observe the elapsed time in each step
 - Like a trace, but with much greater granularity

CICS TASK RECORDER

How is that a new “paradigm”?

- Easy to use
 - Information is displayed in terms familiar to programmer
 - Point-and-click/GUI interface makes usage quick and simple
 - Like a dump, but data is displayed using program variable names
 - Like a trace, but at the level the programmer coded
 - Like a debugger, but avoids the lengthy step-at-a-time process
 - Like a tuning product, in that you can observe delays

CICS TASK RECORDER

How is that a new “paradigm”?

- Provides more information
 - Elapsed time is displayed, showing where tuning might be needed
 - Highlights TCB switches
 - Synchronizes with mirrored transactions in other regions
 - Formats data into the native language variables and structures
 - Provides a way to compare results before and after code changes

CICS TASK RECORDER

How is that a new “paradigm”?

- Speeds application understanding of “legacy” systems
 - Observe transfer of control between programs and across regions
 - See which programs access which VSAM files and Db2 tables
 - Compare changes to data before and after a program call

CICS TASK RECORDER

How do you use CICS Task Recorder?

- Submit a recording session
- Run the transaction(s)
- View the results

USING CICS TASK RECORDER

Submit a recording session

You can start a from either a CICS 3270 session:

```
Session Attributes          Submit a New Session          CICS Task Recorder
Last session has been retrieved
Command ==> ___ (SUB, X=Stop, D=Del, N=New, L=Last)

Tranid ESI* _____
Userid _____ Termid _____ IP _____
Duration 0300 Seconds.  Number of tasks 0300
Record content:  Commarea/Parm Y  VSAM Write/Updt/Del Y  VSAM Read Y
DB Write/Updt/Del Y  Select/Fetch Y  Send/Receive Y  TSQ/TDQ Y  Container Y
Unless next two lines are filled, all recorded tasks are saved.
    Only retain tasks that abended or ran over _____ milliseconds (elapsed)
    Only retain tasks that invoked/accessed program/resource _____
Note _____
Regions: T560 Y T610 N
```

USING CICS TASK RECORDER

...or from a web browser:

Session List **New Session** Users Programs Mappings Running Tasks VSAM-Handy VSAM-Diff TSQ-Handy Db2-Handy Logout

User: WRL1 (Admin) CICS Task Recorder

Which/whose transactions do you wish to record?
Please fill at least one of the fields below.
An asterisk(*) serves as wildcard.

Tranid (up to three):
ESI*

Userid: Myself **Termid:** **IP Address:**

Duration (in seconds): **Number of tasks:**

Unless next lines are filled, all recorded tasks are saved.
Only retain tasks that abended or ran over milliseconds (elapse).
Only retain tasks that invoked/accessed program/resource

Note:

Capture data content for the following commands:

- Commarea / Parm
- VSAM Write/Update/Delete
- VSAM Read
- DB2/Adabas Write/Insert/Update/Delete
- DB2/Adabas Read/Select/Fetch
- Send/Receive
- TS/TD queue
- Container

Participating regions (select up to five):
 T560 T610

Program to trace variables:

Snapshot whole Working-Storage

Variables to trace (select up to five)

- CTR--1
- CTR--2
- CTR--3
- WS-START
- WK-ABSTIME
- WK-LEN
- WK-PGM
- WK-PREV-PGM
- CLEAR-LOW
- S-CURSOR-SET
- S-SEND-FULL
- S-READ-OPT-REC

USING CICS TASK RECORDER

You can view the recording while the session is running, or afterwards by clicking “Tasks”

The screenshot displays the CICS Task Recorder interface. At the top, a navigation bar includes options like 'Session List', 'New Session', 'Users', 'Programs', 'Mappings', 'Running Tasks', and various system options. The 'Running Tasks' section shows a table with columns for 'Select', 'Submitter', 'ID', 'Status', 'Tranid', 'User/Term', 'Start Date', 'Clock', 'End Clock', 'Duration', '#Tasks', and 'Sysid's Note'. A task with ID '1237' and status 'Running' is highlighted. Below the table, a 'Tasks' dropdown menu is open, showing a list of tasks with columns for 'Sysid', 'Started', 'Ended', and 'Mirror'. The task 'T560' is selected, and its details are shown in a pop-up window. This window includes a 'Task counts in each region' table and an 'Other Specifications' table. The 'Task counts in each region' table has columns for 'Sysid', 'Started', 'Ended', and 'Mirror', with values 'T560', '17', '17', and '0' respectively. The 'Other Specifications' table lists various recording options, all of which are set to 'Yes'.

Select	Submitter	ID	Status	Tranid	User/Term	Start Date	Clock	End Clock	Duration	#Tasks	Sysid's Note
Tasks	WRL1	1237	Running	ESI*		2022/01/08	15:40:59	15:45:59	117	17	T560

Sysid	Started	Ended	Mirror
T560	17	17	0

Other Specifications	
Requested duration	300
Number of tasks	300
Program to trace	ESIPADM
Record Commarea / Parm	Yes
Record VSAM Write	Yes
Record VSAM Read	Yes
Record DB2 Write	Yes
Record DB2 Read	Yes
Record Send/Receive	Yes
Record TS/TD queues	Yes
Record Container	Yes

The CICS exits are enabled only for the number of seconds you specify
- No overhead in the region when CTR is not recording.

USING CICS TASK RECORDER

The “Task List” shows the transactions recorded during a session.

Task List Session 1237 Submitter WRL1 Status Time End Started 2022/01/08 15:40:59 Ended 15:45:59 Duration 300 sec. Close CICS Task Recorder

Merge Diff

			Started	Tran	UserID	Elapsed	#Events	Facility	Region	Task#	Notes
<input type="checkbox"/>	Flow	Summary	15:41:49.617939	ESIA	WRL1	11682	116	T03A	T560	600	Next: ESIA Working-Storage
<input type="checkbox"/>	Flow	Summary	15:41:50.847869	ESIA	WRL1	9326	38	T03A	T560	602	CA Len: 512 Next: ESIA Working-Storage
<input type="checkbox"/>	Flow	Summary	15:41:51.349263	ESIA	WRL1	6496	100	T03A	T560	603	CA Len: 512 Key: PF4 Next: ESIA Working-Storage
<input type="checkbox"/>	Flow	Summary	15:41:52.293607	ESIA	WRL1	13556	225	T03A	T560	604	CA Len: 512 Key: PF3 Next: ESIA Working-Storage
<input type="checkbox"/>	Flow	Summary	15:41:52.806238	ESIA	WRL1	10550	166	T03A	T560	605	CA Len: 512 Key: PF3 Next: ESIA Working-Storage
<input type="checkbox"/>	Flow	Summary	15:41:53.453855	ESIA	WRL1	3468	18	T03A	T560	606	CA Len: 512 Key: CLEAR Working-Storage
<input type="checkbox"/>	Flow	Summary	15:41:55.682039	ESIS	WRL1	12625	176	T03A	T560	607	Next: ESIS
<input type="checkbox"/>	Flow	Summary	15:42:18.846200	ESIS	WRL1	5139	41	T03A	T560	608	CA Len: 512 Next: ESIS
<input type="checkbox"/>	Flow	Summary	15:42:19.747843	ESIS	WRL1	11917	120	T03A	T560	609	CA Len: 512 Key: PF6 Next: ESIS\$
<input type="checkbox"/>	Flow	Summary	15:42:19.773431	ESIS\$	WRL1	59089	1800	STARTTRM	T560	610	Key: PF6
<input type="checkbox"/>	Flow	Summary	15:42:25.170159	ESIS\$	WRL1	104488	16788	T03A	T560	611	CA Len: 300 Key: PF3 Next: ESIS@
<input type="checkbox"/>	Flow	Summary	15:42:25.289345	ESIS@	WRL1	7209	43	STARTTRM	T560	612	Key: PF3
<input type="checkbox"/>	Flow	Summary	15:42:27.315547	ESIS	WRL1	33761	380	T03A	T560	613	CA Len: 512 Key: PF9 Next: ESIS
<input type="checkbox"/>	Flow	Summary	15:42:27.357989	ESIL	CICSUSER	7325066	1036	TRANDATA	T560	616	
<input type="checkbox"/>	Flow	Summary	15:42:46.256509	ESIS	WRL1	4539	46	T03A	T560	663	CA Len: 512 Key: CLEAR
<input type="checkbox"/>	Flow	Summary	15:42:47.461973	ESIA	WRL1	7132	116	T03A	T560	664	Next: ESIA Working-Storage
<input type="checkbox"/>	Flow	Summary	15:42:51.455257	ESIA	WRL1	666441	650	T03A	T560	665	Abend ASRA CA Len: 512 Working-Storage

Merge Diff

USING CICS TASK RECORDER

The “Summary” button shows the functions executed in the task.

Summary Task 616: Tranid ESIL, UserID CICSUSER, Started 06:35:57, Sysid T560, Facility TRANDATA, Elapsed 7325066 microsec., In code 30018 Close CICS Task Recorder

Program	Command	Operand	Resp	Count	Elapsed	%Caller	%Task	Code Elapse
ESIPTCP	Socket API	CONNECT		1	1651			
ESIPTCP	WRITEQ TD	CSMT		50	5190			
ESIPTCP	DELAY	ESIDLY1		1	5237K		71.5	
ESIPTCP	READ	ESIFILE		18	44224		00.6	
ESIPTCP	DELETE FILE	ESIFILE		42	202596		02.7	
ESIPTCP	WRITE FILE	ESIFILE		42	122647		01.6	
ESIPTCP	STARTBR	ESIFILE		2	4144			
ESIPTCP	READNEXT	ESIFILE		1	51			
ESIPTCP	ENDBR	ESIFILE		2	105			
ESIPTCP	REWRITE	ESIFILE		2	4308			
ESIPTCP	READ	ESIFILE	notfnd	1	1776			
ESIPTCP	STARTBR	ESIFILE	notfnd	2	5367			
ESIPTCP	ENDBR	ESIFILE	invreq	2	116			
ESIPTCP	READNEXT	ESIFILE	lengerr	1	79			
ESIPTCP	ENQ	ESIFILEQ		3	118			
ESIPTCP	DEQ	ESIFILEQ		4	161			
ESIPTCP	ENQ	ESILENQ1		2	81			
ESIPTCP	READQ TS	ESILSHT1	qiderr	3	232			
ESIPTCP	WRITEQ TS	ESIMON1		1	92			
ESIPTCP	DELETEQ TS	ESIMON1		1	70			
ESIPTCP	INQUIRE TDQUEUE	ESI0	qiderr	1	47			
ESIPTCP	READQ TS	ESILSHT1		1	16			

USING CICS TASK RECORDER

The “Task” button shows the functions executed in the task.

Execution Flow Task 602, Tranid ESIA, UserID WRL1, Started 15:41:50, Sysid T560, Facility T03A, Elapsed 9326 microsec. Close CICS Task Recorder

Next Prev Summary Diff

Expand All Hide Internal Excel Reload Listings

	Program	Offset	Command	Operand	Elps. Before	Elps. During	Accum. Elps.	Notes	Seq
Commarea			Start of task	Sys=T560			0		1
	*CICSRTN		GETMAIN	9024	3126	76	3202		2
	ESIPADM	02B58	COBOL Label	1000-INITIALIZATION			3284		3
WS	ESIPADM	02B94	HANDLE CONDITION		161	180	3543		4
WS ΔWS	ESIPADM	02E14	HANDLE CONDITION		62	37	3643		5
WS ΔWS	ESIPADM	02EA2	RECEIVE MAP	ESIMADM	51	231	3926		6
	ESIPADM	0539E	COBOL Label	9300-CONVERT-MAP-TO-UPPER			3981		7
	ESIPADM	03F9C	COBOL Label	1800-XCTL			4055		8
	ESIPADM	04C26	COBOL Label	8100-PUSH-COMMAREA			4127		9
Item WS ΔWS	ESIPADM	04CAA	WRITEQ TS	ESICT03A	273	151	4350	qiderr	10
Item WS ΔWS	ESIPADM	04D4C	WRITEQ TS	ESICT03A	78	168	4596		11
WS ΔWS	ESIPADM	03FFC	HANDLE CONDITION		74	38	4708		12
Commarea	ESIPADM	0407C	XCTL	ESIPOPT	54		4762		13
	*CICSRTN		FREEMAIN		207	63	5032		14
	*CICSRTN		GETMAIN	11544	216	66	5314		15
	ESIPOPT	03174	COBOL Label	1000-INITIALIZATION			5389		16
	ESIPOPT	031D0	HANDLE CONDITION		117	128	5559		17
	ESIPOPT	0378A	COBOL Label	1200-FIRST-TIME			5588		18
	ESIPOPT	037DA	HANDLE CONDITION		71	37	5666		19


USING CICS TASK RECORDER

Buttons on left show additional information, like Working Storage

Execution Flow Task 602, Tranid ESIA, UserID WRL1, Started 15:41:50, Sysid T560, Facility T03A, Elapsed 9326 microsec. Close CICS Task Recorder

Next Prev Summary Diff

Expand All Hide Internal Excel Reload Listings

	Program	Offset	Command	Operand	Elps. Before	Elps. During	Accum. Elps.	Notes	Seq		
	ESIPADM	03F9C	COBOL Label	1800-XC1L			4055		8		
	ESIPADM	04C26	COBOL Label	8100-PUSH-COMMAREA			4127		9		
Item	WS	ΔWS	ESIPADM	04CAA	WRITEQ TS	ESICT03A	273	151	4350	qiderr	10
Item	WS	ΔWS	ESIPADM	04D4C	WRITEQ TS	ESICT03A	78	168	4596		11
	WS	ΔWS	ESIPADM	03FFC	HANDLE CONDITION		74	38	4708		12
			03 WS-UCTRAN-PGM-REL		17			Char	2	A21	
			02 WS-RELNO					Group	3	A23	
			03 WS-REL		0000			Hexa	2	A24	
			01 TRANSLATE-PARMS					Group	6	A28	
			02 TRANSLATE-LENGTH				0	Binary	4	A28	
			02 TRANSLATE-ACTION		U			Char	1	A2C	
			02 TRANSLATE-RC		00			Hexa	1	A2D	
			01 LENGTH-ESIMADM				353	Binary	2	A30	
			01 ESIMADMI					Group	353	A38	
			02 IO-MAP-TITLEL				0	Binary	2	A44	
			02 IO-MAP-TITLEF		00			Hexa	1	A46	
			02 IO-MAP-TITLEA		00			Hexa	1	A46	
			02 IO-MAP-TITLEI		00000000000000000000000000000000			Hexa	48	A4B	
			02 IO-CUR-DATEL				0	Binary	2	A7B	
			02 IO-CUR-DATEF		00			Hexa	1	A7D	

USING CICS TASK RECORDER

Or the COMMAREA passed to called programs.

The screenshot displays the CICS Task Recorder interface. At the top, it shows the execution flow: Task 602, Trnid ESIA, UserID WRL1, Started 15:41:50, Sysid T560, Facility T03A, Elapsed 9326 microsec. Below this are navigation buttons: Next, Prev, Summary, Diff, and a Command Filter dropdown. There are also buttons for Expand All, Hide Internal, Excel, and Reload Listings.

Item	WS	ΔWS	Program	Offset	Command	Operand	Elps. Before	Elps. During	Accum. Elps.	Notes	Seq
			ESIPADM	04D4C	WRITEQ TS	ESICT03A	78	168	4596		11
			ESIPADM	03FFC	HANDLE CONDITION		74	38	4708		12
			ESIPADM	0407C	XCTL	ESIPOPT	54		4762		13

The detailed view for the selected row (ESIPADM at offset 0407C) shows the following:

- Hex:** A hex editor view showing the data: ESIPADM ESIPADM ...UPDATE ESI.WRL1 WENDELL Lovewell Wendell YY09009.....MY40..... C.....
- Length=512**
- Mapping:** A button that formats the storage into copybooks.
- Diff:** A checkbox for comparing data.

Field Name	Value	Type	Length	Offset
01 WS-COMMAREA		Group	512	00
02 WS-COMM-SYSTEM		Group	256	00
03 WS-COMM-PROGRAM	ESIPADM	Char	8	00
03 WS-COMM-RET-PGM		Group	8	08
04 WS-COMM-RET-PGM-PRE	ESIP	Char	4	08
04 WS-COMM-RET-PGM-SUF	ADM	Char	4	0C
03 WS-COMM-CURR-LEVEL		2 Binary	2	10
03 WS-COMM-CURR-LEVEL-ZD	Hex'0002	Numeric	2	10

The “Mapping” button formats storage into the copybooks used by the program

USING CICS TASK RECORDER

SQL calls can be displayed:

Execution Flow Task 622, Tranid TST7, UserID IB1, Started 11:03:16, Sysid T560, Facility 0003, Elapsed 34937 microsec. Close CICS Task Recorder

Next Prev Summary Diff Expand All Hide Internal Excel Reload Listings

	Program	Offset	Command	Operand	Elps. Before	Elps. During	Accum. Elps.	Notes	Seq
	TST007T	00118	Dynamic call	TST009T	20	258	1468		7
	TST007T	00158	HANDLE ABEND		38	48	1516		11
→	Sql Values	TST007T	0034E	DB2 SQL SELECT 0079	276	2215	4007	TCB Switch	12
	SELECT SSN , NAME , CITY , GENDER , VARCHAR_FORMAT (DOB , 'YYYY/MM/DD') , NUMKIDS , VALUEINT , VALUEFLT INTO : H , : H , : H , : H , : H , : H , : H , : H FROM TESTTFR . PEOPLE WHERE NUMKIDS = : H								
	Sql Values	TST007T	0034E	DB2 SQL SELECT 0079	79	573	4660	SQLCODE=-811	13
→	Sql Values	TST007T	0034E	DB2 SQL SELECT 0079	110	383	5152		14
	SELECT SSN , NAME , CITY , GENDER , VARCHAR_FORMAT (DOB , 'YYYY/MM/DD') , NUMKIDS , VALUEINT , VALUEFLT INTO : H , : H , : H , : H , : H , : H , : H , : H FROM TESTTFR . PEOPLE WHERE NUMKIDS = : H								
	Sql Values	TST007T	00456	DB2 SQL UPDATE 0091	62	8323	13537		15
	Sql Values	TST007T	00530	DB2 SQL UPDATE 0095	65	481	14083	SQLCODE=100	16
	Sql Values	TST007T	0034E	DB2 SQL SELECT 0079	56	384	14522		17
	Sql Values	TST007T	0034E	DB2 SQL SELECT 0079	60	396	14978		18
	Sql Values	TST007T	0034E	DB2 SQL SELECT 0079	59	375	15413	SQLCODE=100	19
	Sql Values	TST007T	0034E	DB2 SQL SELECT 0079	55	332	15800	SQLCODE=100	20
	Sql Values	TST007T	0034E	DB2 SQL SELECT 0079	55	304	16160	SQLCODE=100	21
		TST007T	00580	READQ TS AABBCDDEEFFGGHH	51	285	16495	5 times qiderr	22
	Item	TST007T	005AE	WRITEQ TS AAABBBCC	17	60	16573		23
	Sql Values	TST007T	006EE	DB2 SQL INSERT 0121	45	5080	21698		24

USING CICS TASK RECORDER

...along with their results:

Execution Flow Task 622, Tranid TST7, UserID IB1, Started 11:03:16, Sysid T560, Facility 0003, Elapsed 34937 microsec. Close CICS Task Recorder

Next Prev Summary Diff
Command Filter
Expand All Hide Internal Excel Reload Listings

	Program	Offset	Command	Operand	Elps. Before	Elps. During	Accum. Elps.	Notes	Seq																																								
	TST007T	00118	Dynamic call	TST009T	20	258	1468		7																																								
	TST007T	00158	HANDLE ABEND		38	48	1516		11																																								
→	TST007T	0034E	DB2 SQL SELECT	0079	276	2215	4007	TCB Switch	12																																								
	SELECT SSN , NAME , CITY , GENDER , VARCHAR_FORMAT (DOB , 'YYYY/MM/DD') , NUMKIDS , VALUEINT , VALUEFLT INTO : H , : H , : H , : H , : H , : H , : H , : H FROM TESTTFR . PEOPLE WHERE NUMKIDS = : H																																																
	TST007T	0034E	DB2 SQL SELECT	0079	79	573	4660	SQLCODE=-811	13																																								
	TST007T	0034E	DB2 SQL SELECT	0079	110	383	5152		14																																								
	<input type="checkbox"/> Diff	<table border="1"> <thead> <tr> <th>Type</th> <th>Length</th> <th>Value</th> <th>Value Hexadecimal</th> </tr> </thead> <tbody> <tr> <td>DECIMAL</td> <td>9,0</td> <td>121212121</td> <td>121212121C</td> </tr> <tr> <td>CHAR</td> <td>20</td> <td>Mustafa Cohen</td> <td>D4A4A2A381868140C396888595404040404040</td> </tr> <tr> <td>CHAR</td> <td>20</td> <td>Toronto</td> <td>E396999695A396404040404040404040404040</td> </tr> <tr> <td>CHAR</td> <td>1</td> <td>M</td> <td>D4</td> </tr> <tr> <td>CHAR</td> <td>10</td> <td>1949/03/11</td> <td>F1F9F4F961F0F361F1F1</td> </tr> <tr> <td>SMALLINT</td> <td>2</td> <td>2 0002</td> <td></td> </tr> <tr> <td>LARGEINT</td> <td>4</td> <td>12345858</td> <td>00BC6202</td> </tr> <tr> <td>FLOAT</td> <td>8</td> <td>001112220000.000</td> <td>426F38D4FDF3B646</td> </tr> <tr> <td>SMALLINT</td> <td>2</td> <td>2 0002</td> <td></td> </tr> </tbody> </table>								Type	Length	Value	Value Hexadecimal	DECIMAL	9,0	121212121	121212121C	CHAR	20	Mustafa Cohen	D4A4A2A381868140C396888595404040404040	CHAR	20	Toronto	E396999695A396404040404040404040404040	CHAR	1	M	D4	CHAR	10	1949/03/11	F1F9F4F961F0F361F1F1	SMALLINT	2	2 0002		LARGEINT	4	12345858	00BC6202	FLOAT	8	001112220000.000	426F38D4FDF3B646	SMALLINT	2	2 0002	
Type	Length	Value	Value Hexadecimal																																														
DECIMAL	9,0	121212121	121212121C																																														
CHAR	20	Mustafa Cohen	D4A4A2A381868140C396888595404040404040																																														
CHAR	20	Toronto	E396999695A396404040404040404040404040																																														
CHAR	1	M	D4																																														
CHAR	10	1949/03/11	F1F9F4F961F0F361F1F1																																														
SMALLINT	2	2 0002																																															
LARGEINT	4	12345858	00BC6202																																														
FLOAT	8	001112220000.000	426F38D4FDF3B646																																														
SMALLINT	2	2 0002																																															
	SELECT SSN , NAME , CITY , GENDER , VARCHAR_FORMAT (DOB , 'YYYY/MM/DD') , NUMKIDS , VALUEINT , VALUEFLT																																																

USING CICS TASK RECORDER

Differences between result sets can be compared with the “Diff” function:

Execution Flow Task 622, Tranid TST7, UserID IB1, Started 11:03:16, Sysid T560, Facility 0003, Elapsed 34937 microsec. Close CICS Task Recorder

Next Prev Summary Diff
Expand All Hide Internal Excel Reload Listings

	Program	Offset	Command	Operand	Elps. Before	Elps. During	Accum. Elps.	Notes	Seq
	TST007T	00118	LOAD	TST009T	16	61	1190		6
	TST007T	00118	Dynamic call	TST009T	20	258	1468		7
	TST007T	00158	HANDLE ABEND		38	48	1516		11
Sql Values	TST007T	0034E	DB2 SQL SELECT	0079	276	2215	4007	TCB Switch	12
	<input checked="" type="checkbox"/> Diff								
		Type	Length	Value	Value Hexadecimal				
		DECIMAL	9,0	999977777	999977777C				
		CHAR	20	Bluehorse McNeil	C293A485889699A28540D483D58589934040404				
		CHAR	20	Jerusalem	D18599A4A281938594404040404040404040404				
		CHAR	1	M	D4				
		CHAR	10	1958/08/04	F1F9F5F861F0F861F0F4				
		SMALLINT	2	0	0000				
		LARGEINT	4	12345678	00BC614E				
		FLOAT	8	0087667700000.000	4336CAD4FDF3B646				
		SMALLINT	2	0	0000				
		SELECT SSN , NAME , CITY , GENDER , VARCHAR_FORMAT (DOB , 'YYYY/MM/DD') , NUMKIDS , VALUEINT , VALUEFLT INTO : H , : H , : H , : H , : H , : H , : H , : H FROM TESTTFR . PEOPLE WHERE NUMKIDS = : H							
Sql Values	TST007T	0034E	DB2 SQL SELECT	0079	79	573	4660	SQLCODE=-811	13
Sql Values	TST007T	0034E	DB2 SQL SELECT	0079	110	383	5152		14
	<input checked="" type="checkbox"/> Diff								
		Type	Length	Value	Value Hexadecimal				
		DECIMAL	9,0	121212121	121212121C				
		CHAR	20	Mustafa Cohen	D4A4A2A381868140C396888595404040404040404				

USING CICS TASK RECORDER

Columns highlighted with turquoise indicate differences:

Type	Length	Value1	Value1 Hexadecimal	Value2	Value2 Hexadecimal
DECIMAL	9,0	99997777	99997777C	121212121	121212121C
CHAR	20	Bluehorse McNeil	C293A485889699A28540D483D58589934040404	Mustafa Cohen	D4A4A2A381868140C39688859540404040404
CHAR	20	Jerusalem	D18599A4A2819385944040404040404040404	Toronto	E396999695A39640404040404040404040404
CHAR	1	M	D4	M	D4
CHAR	10	1958/08/04	F1F9F5F861F0F861F0F4	1949/03/11	F1F9F4F961F0F361F1F1
SMALLINT	2	0	0000	2	0002
LARGEINT	4	12345678	00BC614E	12345858	00BC6202
FLOAT	8	0087667700000.000	4336CAD4FDF3B646	0011122200000.000	426F38D4FDF3B646
SMALLINT	2	0	0000	2	0002

“Diff” can also be used on most types of data, including VSAM records, COMMAREAS, and Working Storage variables.

USING CICS TASK RECORDER

Here's an example of displaying a merge of 3 transactions:

Task List Session 1213 Submitter IB1 Status Stopped Started 2021/12/31 11:01:55 Ended 11:15:02 Duration 787 sec. Close CICS Task Recorder

Merge Diff

			Started	Tran	UserID	Elapsed	#Events	Facility	Region	Task#	Notes
<input type="checkbox"/>	Flow	Summary	11:02:04.240327	TST1	CICSUSER	177718	348	0005	T560	497	Variable trace Working-Storage
<input type="checkbox"/>	Flow	Summary	11:02:04.245417	CSMI	CICSUSER	171259	21	MRO	T550	533	CA Len: 74 Originated by TST1 (497)
<input type="checkbox"/>	Flow	Summary	11:02:04.249350	CSMI	CICSUSER	164588	40	MRO	T610	723	CA Len: 74 Originated by CSMI (533)
<input type="checkbox"/>	Flow	Summary	11:02:04.254507	CSMI	CICSUSER	155430	78	MRO	T560	498	CA Len: 74 Originated by CSMI (723)
<input type="checkbox"/>	Flow	Summary	11:02:04.257743	CSMI	CICSUSER	149178	130	MRO	T550	534	CA Len: 74 Originated by CSMI (498)
<input type="checkbox"/>	Flow	Summary	11:02:06.328830	TST5	CICSUSER	9676	26	START	T560	500	Started by TST1 (497)
<input type="checkbox"/>	Flow	Summary	11:02:09.479145	TST5	CICSUSER	7170	26	START	T610	726	Started by TST1 (497)
<input type="checkbox"/>	Flow	Summary	11:02:12.621889	TST5	CICSUSER	6679	26	START	T550	538	Started by TST1 (497)
<input type="checkbox"/>	Flow	Summary	11:02:15.768669	TST5	CICSUSER	5468	27	START	T560	541	Started by TST1 (497)
<input type="checkbox"/>	Flow	Summary	11:02:18.905315	TST5	CICSUSER	5437	26	START	T610	731	Started by TST1 (497)
<input type="checkbox"/>	Flow	Summary	11:02:37.384625	TSTF	CICSUSER	17127	23	0004	T560	548	Key: PF17
<input type="checkbox"/>	Flow	Summary	11:02:45.985763	TST2	IB1	10911	76	0008	T550	542	Variable trace Working-Storage
<input type="checkbox"/>	Flow	Summary	11:02:51.147121	TST3	IB1	14241	51	0006	T610	733	
<input type="checkbox"/>	Flow	Summary	11:02:57.039723	TST5	IB1	6119	37	0003	T560	616	
<input type="checkbox"/>	Flow	Summary	11:03:16.861225	TST7	IB1	34937	57	0003	T560	622	1 TCB switch
<input type="checkbox"/>	Flow	Summary	11:03:20.077856	TST0	IB1	7102	10	0003	T560	624	1 TCB switch
<input checked="" type="checkbox"/>	Flow	Summary	11:03:32.297700	TSTE	CICSUSER	50353083	58	0005	T560	625	
<input checked="" type="checkbox"/>	Flow	Summary	11:03:34.725930	TSTE	CICSUSER	56322182	58	0004	T560	626	
<input checked="" type="checkbox"/>	Flow	Summary	11:03:36.784107	TSTE	IB1	58265846	58	0003	T560	627	
<input type="checkbox"/>	Flow	Summary	11:05:28.703940	TSTS	CICSUSER	28835	32	0005	T560	644	
<input type="checkbox"/>	Flow	Summary	11:05:33.636986	TSTQ	CICSUSER	14725	50	0005	T560	645	

Merge Diff

USING CICS TASK RECORDER

Notice you can see the effects of the ENQs between the different tasks:

3 Concurrent Tasks Merged Side by Side										Close	CICS Task Recorder
Accum. Elapsed	Tran TSTE Task# 625			Tran TSTE Task# 626			Tran TSTE Task# 627				
	Program	Command	Operand	Program	Command	Operand	Program	Command	Operand		
0		Start of task	Sys=T560								
2085	TST00ET	ASSIGN									
2169	TST00ET	HANDLE ABEND									
2350	TST00ET	WRITE OPERATOR	30								
8303	TST00ET	ENQ	A								
8378	TST00ET	SUSPEND									
8444	TST00ET	ENQ	B								
8502	TST00ET	READQ TS	XXXX								
8596	TST00ET	DELAY	TFRTST00								
2428230					Start of task	Sys=T560					
2429851				TST00ET	ASSIGN						
2429901				TST00ET	HANDLE ABEND						
2429999				TST00ET	WRITE OPERATOR	30					
2431941				TST00ET	ENQ	A					
4201679	TST00ET	DEQ	B								
4201838	TST00ET	WRITEQ TS	YYYY								
4202026	TST00ET	DEQ	A								
4202170	TST00ET	ENQ	A								
4202363				TST00ET	SUSPEND						
4202500				TST00ET	ENQ	B					
4202633				TST00ET	READQ TS	XXXX					
4202810				TST00ET	DELAY	TFRTST00					
4486407								Start of task	Sys=T560		
4488967							TST00ET	ASSIGN			
4489026							TST00ET	HANDLE ABEND			
4489112							TST00ET	WRITE OPERATOR	30		
4491054							TST00ET	ENQ	A		
8397976				TST00ET	DEQ	B					
8398122				TST00ET	WRITEQ TS	YYYY					
8398288				TST00ET	DEQ	A					
8398417				TST00ET	ENQ	A					
8398610	TST00ET	SUSPEND									
8398726	TST00ET	ENQ	B								
8398841	TST00ET	READQ TS	XXXX								
8398941	TST00ET	DELAY	TFRTST00								
12589450	TST00ET	DEQ	B								

USING CICS TASK RECORDER

Any questions?

USING CICS TASK RECORDER

Thank you for your time!