



16498: CICS Trace - A Trip into the Unknown

Thomas Dunlap; Themis, Inc.
tomd@themisinc.com

Wednesday – 03/04/2015
8:30 AM Session



SHARE is an independent volunteer-run information technology association that provides **education, professional networking and industry influence.**



Copyright (c) 2014 by SHARE Inc. Except where otherwise noted, this work is licensed under <http://creativecommons.org/licenses/by-nc-sa/3.0/>

Abstract and Agenda

- This session will provide information on how to use trace to solve problems, identify performance issues and profiling of CICS applications.
- Agenda:
 - Overview of trace options
 - Review of trace resources
 - A look at trace formats
 - Using trace for problem resolution
 - Using trace for performance management
 - Using trace for application profiling

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

3/4/2015

- Overview of Trace Options

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

3/4/2015

Overview of Trace Options



- **Standard Trace**

STNTR=OFF – Turn off all of trace
STNTREI=(1,2) – EXEC Interface
STNTRLD=(1) – LD domain
STNTRME=(1,2) – ME domain
STNTRNQ=(1,2) – NQ domain
STNTRRI=(1) – Res Mgr interface
STNTRRM=(1,2) – RM domain
STNTRRX=(1,2) – RX domain
STNTRSM=(1) – SM domain
STNTRXM=(1,2) – XM domain

- **Special Trace**

SPCTR=OFF – Turn off all of trace
SPCTREI=(1,2) – EXEC Interface
SPCTRLD=(1) – LD domain
SPCTRME=(1,2) – ME domain
SPCTRNQ=(1,2) – NQ domain
SPCTRRRI=(1) – Res Mgr interface
SPCTRRM=(1,2) – RM domain
SPCTRRX=(1,2) – RX domain
SPCTRSM=(1) – SM domain
SPCTRXM=(1,2) – XM domain

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



3/4/2015

5

Trace Levels:

Trace points of the latter type each have an associated "level" attribute. Trace levels could, in theory, range in value from 1 to 32, but in practice nearly all mainline trace points have a trace level of 1 or 2. **Level 1** trace points are designed to give enough diagnostic information to resolve "user" errors. These trace entries are produced for the following situations; at **ENTRY** to and **EXIT** from a Domain, at **ENTRY** to and **EXIT** from all major internal Domain functions, *Before* and *After* all calls (**EVENTs**) to other programs/subsystems, and usually at **ENTRY** to and **EXIT** from a component of the AP Domain. **Level 2** trace points are located between the level 1 trace points. They provide information that is likely to be more useful for resolving errors within CICS modules. Level 2 trace points normally would be used only if requested IBM support staff. **Level 3** trace points are well inside of CICS code and *are not documented*, normally IBM internal use only.

Trace Components:

Components are very finite within their implementation and provide tremendous flexibility in controlling the amount of trace output. The components shown on this slide represent the key domain trace activity which is best choice for problem solving with the least amount of trace overhead.


Trace Options:

Trace may also be controlled during CICS initialization through the use of **SIT** or **SIT overrides**. These allow for installation setting of specific standards, but may be changed via **CEMT** or **CETR** transactions; or by **EXEC CICS** commands. The major options are:

SYSTR – Controls the master system trace flag
STNTR – Global specification for standard tracing
TRTABSZ – Size (in kilobytes) of the internal trace table
AUXTR – Auxiliary Trace data sets are to be used
STNTRxx – Standard trace, component specification

USERTR – Controls the master User trace flag
SPCTR – Global specification for special tracing
GTFTTR – MVS GTF is a destination for trace
AUXTRSW – Auxiliary trace autoswitch facility
SPCTRxx – Special trace, component specification

CETR Transaction



CETR
CICS Trace Control Facility
TCA4 TCICSA4


Type in your choices.

Item	Choice	Possible choices
Internal Trace Status	====> STARTED	STArtd, STOpped
Internal Trace Table Size	====> 4096 K	16K - 1048576K
Auxiliary Trace Status	====> STOPPED	STArtd, STOpped, PAused
Auxiliary Trace Dataset	====> A	A, B
Auxiliary Switch Status	====> NO	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

Complete your session evaluations online at www.SHARE.org/Seattle-Eval




CETR provides online control of the CICS trace facility and its activity. Once presented with the first (menu) screen, any of the input fields can be over-typed with the new values that you require. When ENTER is pressed, CETR issues the necessary commands to set the new values and their changes are recorded in the global and local catalogs. These new values will remain in effect until changed again or a *cold start* is performed.

There are two other main functions of CETR, as follows:

- Setting the component trace options
- Setting the transaction and terminal trace options.

The "Component Trace Options" display is produced by pressing PF4 on the main CETR display.


CETR Transaction 

CETR Component Trace Options TCA4 TCICSA4

Over-type where required and press ENTER. PAGE 1 OF 4


Component	Standard	Special
AP	OFF	OFF
BA	OFF	OFF
BM	OFF	OFF
BR	OFF	OFF
CP	OFF	OFF
DC	OFF	OFF
DD	OFF	OFF
DH	OFF	OFF
DM	OFF	OFF
DP	OFF	OFF
DS	1-2	1-2
DU	OFF	OFF
EC	OFF	OFF
EI	1-2	1-2
EJ	OFF	OFF
EM	OFF	OFF
EP	OFF	OFF

PF: 1=Help 3=Quit 7=Back 8=Forward 9=Messages ENTER=Change

Complete your session evaluations online at www.SHARE.org/Seattle-Eval  3/4/2015 7

The "Component Trace Options" display is produced by pressing PF4 on the main CETR display. PF7 and PF8 can be used to scroll backwards and forwards through the list of components. Values can be changed by overtyping and pressing ENTER.

The trace options for a particular component consist of two sets of 32 flags, which support the 32 possible levels. One set gives the active levels of tracing for STANDARD tracing tasks; the other set gives the levels for SPECIAL tracing tasks. The flags are represented as a list of numbers, **1 through 32**, separated by commas. Consecutive values in the list are shortened to a range notation, i.e. 1-3 is the same as 1,2,3. The special values **ALL** and **OFF** are used to represent "all flags on" and "no flags on" respectively.

CETR Transaction 


CETR Component Trace Options TCA4 TCICSA4

Over-type where required and press ENTER. PAGE 4 OF 4

Component	Standard	Special
SJ	OFF	OFF
SM	1-2	1-2
SO	OFF	OFF
ST	OFF	OFF
SZ	OFF	OFF
TC	OFF	OFF
TD	OFF	OFF
TI	OFF	OFF
TR	1-2	1-2
TS	OFF	OFF
UE	OFF	OFF
US	OFF	OFF
WB	OFF	OFF
WU	OFF	OFF
W2	OFF	OFF
XM	1-2	1-2
XS	OFF	OFF


w This is the last page

PF: 1=Help 3=Quit 7=Rack 8=Forward 9=Messages ENTER=Change

Complete your session evaluations online at www.SHARE.org/Seattle-Eval  3/4/2015 8

This is the fourth screen of domains and components. We have not included the two other screens in the interest of saving discussion time. However, any of these options may be changed by simply over-typing the desired domains and / or components and hitting ENTER.

CETR Transaction



CETR
Transaction and Terminal Trace
TCA4 TCICSA4


Type in your choices.

Item	Choice	Possible choices
Transaction ID	====>	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

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



The "Transaction and Terminal Trace" display is produced by pressing PF5 on the main CETR display. It can be used to set STANDARD, SPECIAL, or SUPPRESSED tracing for particular CICS tasks. This is done by setting the appropriate value for the transaction ID used to start the task, or for the terminal from which the task is to be run. *Note that **SUPPRESSED** cannot be set for terminals, only for transaction IDs.* The required terminal can also be specified by using its Netname.

This display can also be used to enable VTAM exit tracing and ZCP trace for specified terminal IDs or netnames. There are also fields that allow enabling of all non terminal VTAM exits, and PF6 can be used to cancel all exit tracing.

CICS Trace Print Utility

Trace print processing overview:

```

* /*JOBPARM LINES=900000
* //AUXTRACE EXEC PGM=DFHTU690,REGION=0M
* //STEPLIB DD DSN=DFH520.CICS.SDFHLOAD,DISP=SHR
* //DFHAUXT DD DSN=TCICSA5.DFHAUXT,DISP=SHR
* /*FHAUXT DD DSN=TCICSA5.DFHBUXT,DISP=SHR
* //DFHAXPRT DD SYSOUT=*
* //DFHAXPRM DD *
*     SHORT or ABBRV or FULL
* /*
* /** EXCEPTION or TIMERG=(hhmmss-hhmmss)

```

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

SHARE
in Seattle 2015

10

This slide shows the utility to print auxiliary trace data. Most sites only have trace captured in memory in dumps. There are times when it might be best to record trace data to a data set.

The diagram shows the process of switching the data sets. From a terminal in CICS (or the console) you enter the transaction CEMT S AUX SW. As the diagram indicates CICS will switch trace data set usage. From data set A to B in this case. *Please be sure to make a note of the now active data set from the response.* You can now use the utility to print the desired dumps.

We have provided a sample set of JCL, although it is modified slightly from the IBM sample. The name of the program is version dependent, DFHTU690 is V5.2. If you have multiple versions just make sure the name matches the version. The rest of the JCL really has not changes for many years.

The DFHAUXT DD will supply the DSN of the trace data set to use. We have DDs for both data sets and just comment one out are required. The rest of the DDs are quite standard for print.

As for the DFHAXPRM statements, we only show the minimum of options. You will need to refer to the IBM documentation for all of the options. What we have here will get you started. The **SHORT** statement specifies that the shorter form of each trace entry should be printed. This will be two lines per entry. The **ABBRV** statement specifies that the abbreviated form of each trace entry should be printed. This will be only one line per entry. The **HULL** statement specifies that the complete form of each trace entry should be printed. This will many lines per entry, some can be a full page of print.

In a JCL comment we show two of the key selection parameters. The **EXCEPTION** specifies that only exception trace entries should be printed. The **TIMERG** specifies the range of time, in hours minutes and second, you want to print trace entries from.

CICS Trace Print Utility



- Parameters for the utility:
 - SHORT, ABBREV, or FULL
 - TRANID=(xxxx,yyyy) – one or more transaction codes
 - TERMID=(xxxx,yyyy) – one or more terminal identifiers
 - TASKID=(nnnnn,nnnnn) – one or more task numbers
 - KE_NUM=(xxxx,xxxx) – one or more KE entry hexadecimal values
 - ENTRY_NUM=(nnnnnn–nnnnnn) – specify a range of entry numbers
 - TIMERG=(hh:mm–hh:mm) – time range of trace entries
 - PAGESIZE=60 – specifies the number of lines on a page
 - UPPERCASE – specifies that output is to be upper case only
 - ALL (default) or EXCEPTION or TYPETR=(ddxxxx,ddxxxx-xxxx)
- Required is one of SHORT, ABBREV, and FULL (default)
- Selection options may refer to a single value, multiple values, or a range of values

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



11

ABBREV|SHORT|FULL

Specifies how much of each trace entry you want printed. If you specify this statement, it must always be the first statement in either the PARM parameter or the DFHAXPRM data set.

ABBREV – Indicates that you require the abbreviated, one-line-per-entry, form of trace print.

SHORT – Indicates that you require the short formatted print of the data in each entry. This consists of the information in the abbreviated format entry, and the following elements from the interpretation string of the fully formatted entry; interpreted parameter list, Return address, Time, and Interval.

FULL – Indicates that you want a fully formatted print of all the data in each entry, (default value).

TRANID=(tttt[,tttt[,,...]]) - This statement specifies the transaction identifiers of one or more transactions for which trace entries are to be printed.

TERMID=(tttt[,tttt[,,...]]) - This statement specifies the terminal identifiers (tttt) of one or more terminals for which trace entries are to be printed.

TASKID=({id | id-id}[,,{id | id-id}[,,...]]) - This statement specifies the task identifiers (id) of one or more tasks for which trace entries are to be printed. An id value can be any number up to five decimal digits long, any of the character strings JAS, J01 through J99, III, TCP, or DSTCB, or any non-numeric two-character domain ID of the attaching domain.

KE_NUM=(xxxx[,xxxx[,,...]]) - This statement specifies that only the entries for tasks with the specified hexadecimal kernel task numbers are printed.

ENTRY_NUM=({nnnnnn | nnnnnn–nnnnnn}[,{nnnnnn | nnnnnn}[,,...]]) - This statement specifies the sequence numbers of one or more trace entries that you want to print. 6 digits each.

CICS Trace Print Utility



- Parameters for the utility:
 - SHORT, ABBREV, or FULL
 - TRANID=(xxxx,yyyy) – one or more transaction codes
 - TERMID=(xxxx,yyyy) – one or more terminal identifiers
 - TASKID=(nnnnn,nnnnn) – one or more task numbers
 - KE_NUM=(xxxx,xxxx) – one or more KE entry hexadecimal values
 - ENTRY_NUM=(nnnnn–nnnnn) – specify a range of entry numbers
 - TIMERG=(hh:mm–hh:mm) – time range of trace entries
 - PAGESIZE=60 – specifies the number of lines on a page
 - UPPERCASE – specifies that output is to be upper case only
 - ALL (default) or EXCEPTION or TYPETR=(ddxxxx,ddxxxx-xxxx)
- Required is one of SHORT, ABBREV, and FULL (default)
- Selection options may refer to a single value, multiple values, or a range of values

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



12

Trace print options (continued):

TIMERG=(hhmmss-hhmmss[,hhmmss-hhmmss,,,,]) - This statement specifies the time period or periods for which trace entries are to be printed.

PAGESIZE=(value) – Specifies the number of lines printed on a page, default is 55.

UPPERCASE – Specifies that you want the trace output in uppercase only.

ALL - Specifies that all trace entries in the auxiliary trace data set are to be printed, default value.

EXCEPTION – Specifies that only exception trace entries in the auxiliary trace data set are to be printed.

TYPETR={({ddxxxx|ddxxxx-xxxx}[, {ddxxxx|ddxxxx-xxxx}]) - This statement specifies the trace entry identifiers for the particular domain entries, specified by the domain or component id and a point id (hexadecimal value) within the domain.

- Overview of Trace Formats

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

3/4/2015

Trace Formats

Abbreviated trace entries "old" and "new" style:

New style:
00051 QR AP E160 EXEC ENTRY READNEXT 'THMFLK1 ' AT X'2950A7F0', AT X'2950FC34', 256 AT X'2950F9E3', '004004' =023445=

(5)
Trace Interpretation

Old style:
00051 QR AP 00E1 E1P ENTRY READNEXT

0004,2950A658 .&w.,0900060E0200 =023444=

(7)
Sequence no.

(1)
Task no. and TCB

(2)
Trace Point

(3)
Module

(4)
Trace Type

(5)
Trace Interpretation

(6)
Request/Response, Field A, Field B, Resource

Single line of output printed in dump or auxiliary trace print

Minimal information provided


Designed to be easy to find key information

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

3/4/2015
14

The abbreviated trace entries are a single line of output, providing the minimal amount of information. These type of entries will show in transaction dump output, system dump (IPCS) output on request, and the auxiliary trace print utility (DFHTU690) on request.

- (1) **Task identification number** or special id for certain AP domain system transactions (e.g., **TCB**). Also included in the first trace area is the TCB the task is dispatched on.
- (2) Indicates the originating place, *trace point*, where the event occurred. These do have some documentation provided in the CICS TS Trace Entries manual.
- (3) Identifies the **module in the AP domain** or **gate call being performed** where the event (code) being traced is located.
- (4) May be **ENTRY, EXIT, EVENT** or **DATA** specifying whether a call is being traced or a significant happening transpired or data exists which might be of interest. For each ENTRY there will be a corresponding EXIT trace entry. These can be used to establish exactly which module the task is in at that point in time.
- (5) Short "**English-like**" **explanation** of the contents of the trace entry consisting of request & options and other parameters
- (6) Coded *request, request dependent data fields* and *resource* without interpretation as found in Full Trace. Some requests name a particular associated resource, *not present on most entries*.
- (7) An identifying number in order to **relate abbreviated and full trace** entries to one another.



Trace Formats

Full trace entries "old" and "new" style:

New style:

```
AP E160 EXEC ENTRY READNEXT FILE('THMFLK1 ' AT X'2950A7F0') INTO( AT X'2950FC34') LENGTH(256 AT X'2950F9E3') RIDFLD('004004' AT X'2950F97C') KEVLENGTH(6 AT X'2950F014') REQID(0 AT X'A950A7F8') NOHANDLE COBOLII STMT_#(00372)
```

```
TASK-00051 KE_NUM-006B TCB-QR /008E68F8 RET-800864BE TIME-14:55:44.2698538908 INTERVAL-00.0000457812 =023445=
1-0000 00780000 00162950 A7D80000 0000060E FC002700 008000F0 F0F3F7F2 00010110 .....&xQ.....00372...
0020 2950A7F0 5046CF0 E3C804CG D3D2F140 0002A008 2950FC34 A950F9A2 0103030A ..&x0&..0THMFLK1.....&.z69s...
0040 2950F9E3 050AA950 01000004 040E2950 F97C030A 2950F0F0 F4F0F0F4 0105050A ..69T.z&.....698...6004004...
0060 2950F014 0407C7D4 00060106 060AA950 A7F84040 40400000 .....&0.MPGM.....z&x8 ..
2-0000 2950A7D8 2950A7F0 2950FC34 2950F9E3 2950F97C 2950F014 A950A7F8 ..&xQ.&x0.&...69T.698.60.z&x8 *
3-0000 2950A7B8 .....&x.....
4-0000 0145543C 0114162F E3C804F6 0000051C 00000000 00000000 00000006 0C000000 .....THM6.....
0020 000000E3 C8D4C6D3 D2F14000 00000000 000000E3 C8D4C6D3 D2F14000 00000000 .....THMFLK1.....THMFLK1.....
0040 00000000 00000000 00000000 00000000 00000000 00 .....
5-0000 2950A658 .....&w.....
6-0000 00104001 2950A480 00000000 A9421036 00000000 2950B998 2950A768 29500100 .....&u.....z.....&.q.&x.&.
0020 2941C150 29500820 2950FBA0 2950A7F8 29512BA0 2950EBA0 29508848 2941B150 ..&.A&.&...&x&.....&.....&
0040 2942016E 29509780 .....&p.....
```

Old style:

```
AP 00E1 EIP ENTRY READNEXT .....REQ(0004) FIELD-A(2950A658 .&w.) FIELD-B(0900060E ....) BOUNDARY(0200)
```

```
TASK-00051 KE_NUM-006B TCB-QR /008E68F8 RET-A9421036 TIME-14:55:44.2698081096 INTERVAL-00.0000360025 =023444=
```

(1)
Trace Point,
Module,
Trace Type

(2)
Second
common line
of output


(3)
Trace
Interpretation

(4)
Request/Response,
Field A,
Field B,
Resource

(5)
Sequence
no.


Multiple line output, maximum amount of information

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

 3/4/2015 15

The full trace entries are a single line of output, providing the minimal amount of information. These type of entries will show in transaction dump output, system dump (IPCS) output on request, and the auxiliary trace print utility (DFHTU690) on request.

- (1) Indicates the trace point, module name or gate call, plus the trace entry type. The trace points do have some documentation provided in the CICS TS Trace Entries manual. The module names are relative to modules in the AP domain. The gate calls show the function of a domain is being invoked. The trace entry type May be **ENTRY**, **EXIT**, **EVENT**, **CALL**, **RETUR**, ***EXC*** or **DATA** specifying whether a call is being traced or a significant happening transpired or data exists which might be of interest. **For each command performed by an application program there will be and ENTRY & EXIT combination.** The EIP trace type entries can be used to provide a profile of the commands executed by the application.
- (2) The second line of the trace entry contains several key pieces of information. The first (TASK=) identifies the task number, component of the AP domain or CICS domain with which the entry is associated. This is followed by the KE domain task number assigned plus the TCB address. This information can provide a tie back to the KE summaries and Kernel Error data. Next id the return address (RET-) or Reg. 14, to the module which has issued the call. When the entries show a module of EIP then the return address can be tracked to the application call statement for the CICS command. Next is the time at which the trace entry was recorded with a very small fraction of a second in time. Last the INTERVAL is the amount of time between this trace entry and the previous entry. This can at times indicate a potential performance issue when the time interval is considered too large (asterisk (*)) when over 50 ms).
- (3) Short "**English-like**" explanation of the contents of the trace entry request type, with pertinent qualifying information. However, the new style trace entries contain several keyword(value) pairs which provide a very detailed description of the attributes for an CICS command.



Trace Formats

Full trace entries “old” and “new” style:

New style:

```
AP E160 EXEC ENTRY READNEXT FILE('THMFLK1 ' AT X'2950A7F0') INTO( AT X'2950FC34') LENGTH(256 AT X'2950F9E3') RIDFLD('004004' AT X'2950F97C') KEYLENGTH(6 AT X'2950F014') REQID(0 AT X'A950A7F8') NOHANDLE COBOLII STMT_#(00372)
```

```
TASK-00051 KE_NUM-006B TCB-OR /008E68F8 RET-800864BE TIME-14:55:44.2698538908 INTERVAL-00.0000457812 =023445=
1-0000 00780000 00162950 A7D80000 0000060E FC002700 008000F0 F0F3F7F2 00010110 .....&xQ.....00372....*
0020 2950A7F0 5046ECF0 E3C804C6 D3D2F140 0002A008 2950FC34 A950F9A2 0103030A ..&x0&..0THMFLK1.....&..z69s....*
0040 2950F9E3 050AA950 01000004 040E2950 F97C030A 2950F0F0 F4F0F0F4 0105050A ..&9T..z&.....&98...&004004....*
0060 2950F014 0407C7D4 00060106 060AA950 A7F84040 40400000 .....&0.MPGM.....z&x8 ..*
2-0000 2950A7D8 2950A7F0 2950FC34 2950F9E3 2950F97C 2950F014 A950A7F8 ..&xQ.&x0.&...&9T.&98.&0.z&x8 ..*
3-0000 2950A7B8 .....&x*
4-0000 0145543C 0114162F E3C804F6 0000051C 00000000 00000000 00000006 0C000000 .....THM6.....*
0020 000000E3 C8D4C6D3 D2F14000 00000000 000000E3 C8D4C6D3 D2F14000 00000000 .....THMFLK1.....*
0040 00000000 00000000 00000000 00000000 00000000 00 .....*
5-0000 2950A658 .....&w.....*
6-0000 00104001 2950A480 00000000 A9421036 00000000 2950B998 2950A768 29500100 ...&u.....z.....&q.&x.&.....*
0020 2941C150 29500820 2950FBA0 2950A7F8 295128A0 2950E8A0 29508848 2941B150 ..&A&.&...&.&x&.....&.....&*
0040 2942016E 29509780 .....&p.....*
```

Old style:

```
AP 00E1 EIP ENTRY READNEXT REQ(0004) FIELD-A(2950A658 .&w.) FIELD-B(0900060E ....) BOUNDARY(0200)
```

```
TASK-00051 KE_NUM-006B TCB-OR /008E68F8 RET-A9421036 TIME-14:55:44.2698081096 INTERVAL-00.0000306025 =023444=
```

(1)
Trace Point,
Module,
Trace Type

(2)
Second
common line
of output


(3)
Trace
Interpretation

(4)
Request/Response,
Field A,
Field B,
Resource

(5)
Sequence
no.

Multiple line output, maximum amount of information

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

 3/4/2015 16

Full Trace Format (continued):

- (4) **REQ()** – Combination of bit settings which indicate what type of request/response is being recorded in which AP domain component.
 - FIELD-A()** and **FIELD-B()** – Interpretation totally dependent upon the request type and trace point involved. The User Handbook and Problem Determination Manual will contain a description of the information found in FIELD-A and FIELD-B. However for the **AP 00E1** entries which we are most concerned about, contain the following information:
 - For ENTRY** – FIELD-A contains **address of the program RSA**; which is the DSA of PL/I or C, the TGT of COBOL and DFHEISTG of assembler. FIELD-B contains the execution key in the first byte, plus **EIBFN** or function code in the last two bytes.
 - For EXIT** – FIELD-A contains the same value as found in **EIBRESP2**, and FIELD-B contains the **EIBRESP** value in the first two bytes, plus **EIBFN** in the last two bytes,
 - RESOURCE()** – Some requests name a particular resource which the component is manipulating identified by the **RESOURCE()** keyword. Depending upon the trace point involved, this may represent a file name, TS or TD queue name, a terminal identifier, etc.
- (5) An identifying number in order to relate abbreviated and full trace entries to one another. *Not an actual part of the trace entry, but rather a number assigned during the printing of the trace entries.*
- (6) Each “New Style Full Trace” contains many data areas associated with the trace entry. These are somewhat documented as part of the trace point documentation found within the CICS TS Trace Entries manual.

Trace Formats

Abbreviated "exception" trace entries "old" and "new" style:

New style:
 00113 QR AP 1942 APLI *EXC* Runaway START_PROGRAM,THMPGMC,CEDF,FULLAPI,EXEC,NO,280CC9FC,00000000 , 000000 =025966=

Old style:
 00113 QR AP 079Z SRP *EXC* RUNAWAY_TASK =025967=

(1) Task no. and TCB
 (2) Trace Point
 (3) Module
 (4) Trace Type
 (5) Trace Interpretation
 (6) Sequence no.

Single line of output printed in dump or auxiliary trace print


Minimal information provided

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

SHARE in Seattle 2015
 3/4/2015 17

The abbreviated exception trace entries are a single line of output, providing the minimal amount of information. These type of entries will show in transaction dump output, system dump (IPCS) output, and the auxiliary trace print utility (DFHTU690) at all times.

- (1) **Task identification number** or special id for certain AP domain system transactions (e.g., **TCP**). Also included in the first trace area is the TCB the task is dispatched on.
- (2) Indicates the originating place, *trace point*, where the event occurred. These do have some documentation provided in the CICS TS Trace Entries manual.
- (3) Identifies the **module in the AP domain** or **gate call being performed** where the event (code) being traced is located.
- (4) May be ***EXC*** or ***EXCU** specifying whether the exception is CICS or USER generated, respectively.
- (5) Short "**English-like**" **explanation** of the contents of the trace entry consisting of request & options and other parameters
- (6) An identifying number in order to **relate abbreviated and full trace** entries to one another.



Trace Formats

Full "exception" trace entries "old" and "new" style:

New Style:
 AP 1942 APLI *EXC* - Runaway FUNCTION(START_PROGRAM) PROGRAM(TMHPGMC) CEDF_STATUS(CEDF) EXECUTION_SET(FULLAPI)
 ENVIRONMENT_TYPE(EXEC) SYNCONRETURN(NO) LANGUAGE_BLOCK(280CC9FC) COMMAREA(00000000 , 00000000) LINK_LEVEL(2)
 SYSEIB_REQUEST(NO)

```

TASK-00113 KE_NUM-004A TCB-OR /008E68F8 RET-ASC01732 TIME-15:20:06.1075540471 INTERVAL-00.0000975312 =025966=
1-0000 01880000 0000000A 00000000 00000000 88184750 00000000 02000100 E3C8D4D7 *.....&.....THMP*
0020 C7d4c340 00000000 00000000 280870FD 00000001 01410202 29418000 280CC9FC *GMC.....0.....I.*
0040 00008858 00008090 00000000 A670088c 00000000 00000000 00000000 00029102 *.....w.....j.*
      *.....2%.....3}.....{.....*
      *.....r}.....0.....i.....*
      *---/AKEC.....DFHAPLII......SI.*
      *.....rj.....j.....oH.....*
      *.....k.....*
      *.....*
0180 00000000 2684F26C 00000000 2684F3D0 00000000 26CF9CC0 00000000 27306230
01A0 00000000 2599D100 00000000 2684DEFD 00000000 255A9D9C
2-0000 60606061 C1D2C5C3 03000000 00000000 C4C6C8C1 D7D3C9F1 00000000 25F8C900
0020 00000000 2599D100 00000000 26849100 3596C800 00000017 00000000 FFFFFFFF
0040 00000000 80000000 00000000 000864BE 00000000 00000000 00000000 00000000
      *.....k.....*
      *.....*
02A0 00000000 80000000 00000000 00000000 00000000 00000000 25739200 00000000
02C0 00000000 00000000 00000000 00000000
  
```

Old style:
 AP 0792 SRP *EXC* - RUNAWAY_TASK
 TASK-00113 KE_NUM-004A TCB-OR /008E68F8 RET-A598A3FC TIME-15:20:06.1075893283 INTERVAL-00.0000352812 =025967=

(5) Trace Detail

(1) Trace Point, Module, Trace Type


(2) Second common line of output

(3) Trace Interpretation

(4) Sequence no.

Multiple line output, maximum amount of information


Complete your session evaluations online at www.SHARE.org/Seattle-Eval



3/4/2015 18

The full exception trace entries are a single line of output, providing the minimal amount of information. These type of entries will show in transaction dump output, system dump (IPCS) output, and the auxiliary trace print utility (DFHTU690) at all times.

- (1) Indicates the trace point, module name or gate call, plus the trace entry type. The trace points do have some documentation provided in the CICS TS Trace Entries manual. The module names are relative to modules in the AP domain. The gate calls show the function of a domain is being invoked. The trace entry type May be ***EXC*** or ***EXCU** specifying whether the exception is CICS or USER generated, respectively.
- (2) The second line of the trace entry contains several key pieces of information. The first (TASK=) identifies the task number, component of the AP domain or CICS domain with which the entry is associated. This is followed by the KE domain task number assigned plus the TCB address. This information can provide a tie back to the KE summaries and Kernel Error data. Next is the return address (RET-) or Reg. 14, to the module which has issued the call. When the entries show a module of EIP then the return address can be tracked to the application call statement for the CICS command. Next is the time at which the trace entry was recorded with a very small fraction of a second in time. Last the INTERVAL is the amount of time between this trace entry and the previous entry. This can at times indicate a potential performance issue when the time interval is considered too large (an asterisk (*) when over 50 ms).
- (3) Short **"English-like" explanation** of the contents of the trace entry request type, with pertinent qualifying information. However, the new style trace entries contain several keyword(value) pairs which provide a very detailed description of the attributes about why the exception occurs.
- (4) An identifying number in order to relate abbreviated and full trace entries to one another. *Not an actual part of the trace entry, but rather a number assigned during the printing of the trace entries.*
- (5) Each "New Style Full Trace" contains many data areas associated with the trace entry. These are somewhat documented as part of the trace point documentation found within the CICS TS Trace Entries manual.



EIP Trace Entries

Comparison between ABBREV, SHORT, and FULL trace entries from auxiliary trace output

```


ABBREV Style:
00113 QR AP 00E1 EIP ENTRY READQ-TS 0004,29513DF8 ...8,09000A04 .....0200 =024716=
00113 QR AP E160 EXEC ENTRY READQ TS 'THEMISTX' AT X'295158B8',AT X'29515BC0',32 AT X'29515C41',1 AT X'

SHORT Style:
00113 QR AP 00E1 EIP ENTRY READQ-TS REQ(0004) FIELD-A(29513DF8 ...8) FIELD-B(09000A04 ....)
BOUNDARY(0200) RET-A9425222 15:20:06.0116540158 00.0000053906 =024716=
00113 QR AP E160 EXEC ENTRY READQ TS QUEUE('THEMISTX' AT X'295158B8') INTO( AT X'29515BC0') LENGTH(32 AT X'29515C41') ITEM(1 AT
X'A9515280') NOHANDLE COBOLII STMT_#(00017)
RET-8008648E 15:20:06.0116639377 00.0000099218 =024717=

FULL Style:
AP 00E1 EIP ENTRY READQ-TS REQ(0004) FIELD-A(29513DF8 ...8) FIELD-B(09000A04 ....) BOUNDARY(0200)
TASK-00113 KE_NUM-004A TCB-QR /008E68F8 RET-A9425222 TIME-15:20:06.0116540158 INTERVAL-00.0000053906 =024716=
AP E160 EXEC ENTRY READQ TS QUEUE('THEMISTX' AT X'295158B8') INTO( AT X'29515BC0') LENGTH(32 AT X'29515C41') ITEM(1 AT X'A9515280')
NOHANDLE COBOLII STMT_#(00017)
TASK-00113 KE_NUM-004A TCB-QR /008E68F8 RET-8008648E TIME-15:20:06.0116639377 INTERVAL-00.0000099218 =024717=
1-0000 00580000 00162951 3F180000 00000A04 E8002700 008900F0 F9F0F1F7 00019910 .....Y.....i.00017..F.*
0020 295158B8 29515C08 E3C8C504 C9E2E3E7 0002A008 295158C0 C50840C6 0103030A .....THEMISTX.....$(EQ F....*
0040 29515C41 40407E5C 00200105 2D0AA951 52B0515C 00200001 .....=.....Z.....*
2-0000 29513F18 295158B8 29515BC0 29515C41 29513F30 A9515280 .....S{...Z...*
3-0000 29513F00 .....*
4-0000 0152006F 0114162F E3C8D4F4 0000113C D7F0F2F7 00000004 00007D12 04000000 .....?.....THM4.....P027.....*
0020 00000000 00000000 00000000 00000000 00000040 40404040 40404000 00000000 .....*
0040 00000000 00000000 00000000 00000000 00000000 00 .....*
5-0000 29513DF8 .....*
6-0000 00104001 29513C50 00000000 A9425222 00000000 29515144 29513F00 29515280 .....&.....Z.....*
0020 00000000 29500100 2684F5A8 2684F560 00000000 29515280 29514FE8 29423C9C .....&.....5y..5.....|Y....*
0040 294251C0 29512F20 .....*

```

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



3/4/2015 19

The slide shows a comparison between the three different formats of the trace entries output from the trace print utility, DFHTU690. The same two trace entries are shown in the different formats to provide a good comparison. As you can see, the **abbreviated format** is a single line of output with only the minimal of information. The **short format** is 1, 2, or 3 lines of output from the utility. Included are some **keyword(value)** attributes, plus the inclusion of the **RET-** and **time fields**. The latter being something we have wanted for quite awhile on the abbreviated entries. Last are the **full formatted** entries, which contain not only the same information as the short entries but also several data areas. The data areas provided are somewhat documented and are deemed relevant to the entry produced.


If you will notice, these entries represent the fact that an application has issued a **READQ TS** command. These are produced at “entry” to the command level interface from the application program. All of the information represents the command which CICS is about to execute.

The **AP 00E1 EIP** entry is the result of EI trace level 1 and the **AP E160 EXEC** entry is the result of EI trace level 2. The AP 00E1 shows the minimal of information. However, the AP E160 entry includes the important information like the name of the TS queue, the **LENGTH** of the item and its number (**ITEM**), plus the COBOL statement number (**COBOL_STMT_#**).

The **AP 00E1** short and full entries both provide the address of the command within the program, **RET-** attribute. This can be useful in providing a profile of the commands being executed by the application, especially if you do not have the source code. This can also get you back to the logic in the source program to more easily resolve problems. Please do not use the attribute on the AP E160 entries as this relates to a CICS module within the AP domain normally. RET-

On the **AP E160** full trace entry, some key data areas are #4 – the EIB at this point in time, #5 – the address of the RSA (register save area), and #6 – actual RSA.

EIP Trace Entries



Comparison between ABBREVV, SHORT, and FULL trace entries from auxiliary trace output

```

ABBREVV Style:
00113 QR   AP  E161 EXEC  EXIT  READQ           TS 'THEMISTX' AT X'295158B8', '>= THEMIS TSQ RECORD FOR AICA =<' AT X' =024720=
00113 QR   AP  00E1 EIP   EXIT  READQ-TS      OK                        00F4,00000000 .....00000A04 .....0200 =024721=

SHORT Style:
00113 QR   AP  E161 EXEC  EXIT  READQ TS QUEUE('THEMISTX' AT X'295158B8') INTO('>= THEMIS TSQ RECORD FOR AICA =<' AT X'295158C0')
                                LENGTH(32 AT X'29515C41') ITEM(1 AT X'A95152B0') RESP(0) RESP2(0) NOHANDLE COBOLII STMT_#
                                (00017)                                RET=800885A4 15:20:06.0117363908 00.0000193593 =024720=
00113 QR   AP  00E1 EIP   EXIT  READQ-TS OK      REQ(00F4) FIELD-A(00000000 ..... ) FIELD-B(00000A04 ..... )
                                BOUNDARY(0200)                               RET=A9425222 15:20:06.0117385471 00.0000021562 =024721=

FULL Style:
AP  E161 EXEC EXIT READQ TS QUEUE('THEMISTX' AT X'295158B8') INTO('>= THEMIS TSQ RECORD FOR AICA =<' AT X'295158C0') LENGTH(32 AT
X'29515C41') ITEM(1 AT X'A95152B0') RESP(0) RESP2(0) NOHANDLE COBOLII STMT_#(00017)


TASK-00113 KE_NUM-004A TCB-QR /008E68F8 RET=800885A4 TIME=15:20:06.0117363908 INTERVAL=00.0000193593 =024720=
1-0000 00780000 00162951 3F180000 00000A04 E8002700 008900F0 F0F0F1F7 00019910 *.....Y...i.00017..F.*
0020 295158B8 29515C08 E3C8C5D4 C9E2E3E7 0002A028 295158C0 C50840C6 6E7E40E3 *..$.$.THEMISTX.....$[EQ F=> T*
0040 C8C5D4C9 E240E3E2 D84009C5 C30609C4 40C60609 40C1C9C3 C1407E4C 0103030A *HEMIS TSQ RECORD FOR AICA =<...$*
0060 29515C41 03000F27 00200105 200AA951 32B0F57A F2F00001 *.....$.$.5{...$.5:20...*
2-0000 29513F18 295158B8 295158C0 29515C41 29513F30 A95152B0 *.....*
3-0000 29513F00 *.....*
4-0000 0152006F 0114162F E3C804F4 0000113C D7F0F2F7 00000004 00007D0A 04000000 *..?....THM4...P027.....*
0020 00000000 00000000 00000000 00000000 000000E3 C8C5D4C9 E2E3E700 00000000 *.....THEMISTX.....*
0040 00000000 00000000 00000000 00000000 00000000 00 *.....*
5-0000 29513DF8 *.....8*
6-0000 00104901 29513C50 00000000 A9425222 00000000 29515144 29513F00 295152B0 *.....&...z.....*
0020 00000000 29500100 26B4F5A8 26B4F560 00000000 295152B0 29514FE8 29423C9C *.....&...5y..5.....|Y...*
0040 294251C0 29512F20 *.....f.....*

AP  00E1 EIP EXIT READQ-TS OK                                REQ(00F4) FIELD-A(00000000 ..... ) FIELD-B(00000A04 ..... ) BOUNDARY(0200)

TASK-00113 KE_NUM-004A TCB-QR /008E68F8 RET=A9425222 TIME=15:20:06.0117385471 INTERVAL=00.0000021562 =024721=

```

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



3/4/2015 20

The slide shows a comparison between the three different formats of the trace entries output from the trace print utility, DFHTU690. The same two trace entries are shown in the different formats to provide a good comparison. As you can see, the **abbreviated format** is a single line of output with only the minimal of information. The **short format** is 1, 2, or 3 lines of output from the utility. Included are some keyword(value) attributes, plus the inclusion of the **RET-** and **time fields**. The latter being something we have wanted for quite awhile on the abbreviated entries. Last are the **full formatted** entries, which contain not only the same information as the short entries but also several data areas. The data areas provided are somewhat documented and are deemed relevant to the entry produced.

If you will notice, these entries represent the fact that a **READQ TS** command has been done. These are produced at “exit” from the command level interface returning to the application program. All of the information represents the command which CICS has just completed.

The **AP 00E1 EIP** entry is the result of EI trace level 1 and the **AP E161 EXEC** entry is the result of EI trace level 2. The AP 00E1 shows the minimal of information. However, the AP E161 entry includes the important information like the name of the TS queue, the **LENGTH** of the item and its number (**ITEM**), plus the COBOL statement number (**COBOL_STMT_#**). Since this entry is the result of a completed command, the **RESP()** and **RESP2()** attributes are included to provide the status of command completion.

The **AP 00E1** short and full entries both provide the address of the command within the program, **RET-** attribute. This can be useful in providing a profile of the commands being executed by the application, especially if you do not have the source code. This can also get you back to the logic in the source program to more easily resolve problems. Please do not use the **RET-** attribute on the AP E161 entries as this relates to a CICS module within the AP domain normally.

On the **AP E161** full trace entry, some key data areas are #4 – the EIB at this point in time, #5 – the address of the RSA (register save area), and #6 – actual RSA.

CICS Command Entry with UE Enabled



EXEC CICS command trace with inclusion of user exit trace entries:

```
00054 QR AP 00E1 EIP ENTRY PUT-CONTAINER BOUNDARY(0200) REQ(0004) FIELD-A(27D0A698 )HQ) FIELD-B(09003416 .....)  
00054 QR AP E110 EISR ENTRY TRACE_ENTRY PARM_ADDRESS(27D0A7D0) RET-A7419026 12:38:08.6320230471 00.0000123593 =016048=  
00054 QR AP E160 EXEC ENTRY PUT_CONTAINER('THEMISCONT' AT X'27D0FA20') CHANNEL('THEMISCHNL' AT X'27D0FA10') FROM( RET-8008661E 12:38:08.6320280627 00.0000050156 =016049=  
(00049) AT X'27D0F8D0') FLENGTH(32 AT X'27D0FA30') DATATYPE(1019 AT X'A7D0A960') COBOLII STMT_# RET-8008661E 12:38:08.6320680627 00.0000400000 =016050=  
00054 QR AP E111 EISR EXIT TRACE_ENTRY/OK RET-8008661E 12:38:08.6320712502 00.0000031875 =016051=  
00054 QR AP D500 UEH EVENT LINK-TO-USER-EXIT-PROGRAM EQAXEIN AT EXIT POINT XEIN RET-80086A06 12:38:08.6320865939 00.0000153437 =016052=  
00054 QR AP D501 UEH EVENT RETURN-FROM-USER-EXIT-PROGRAM EQAXFTIN WITH RETURN CODE 0 RET-80086A06 12:38:08.6320922658 00.0000056718 =016053=  
00054 QR AP D500 UEH EVENT LINK-TO-USER-EXIT-PROGRAM EQANCXIN AT EXIT POINT XEIN RET-80086A06 12:38:08.6320972033 00.0000049375 =016054=  
00054 QR AP D501 UEH EVENT RETURN-FROM-USER-EXIT-PROGRAM EQANCXIN WITH RETURN CODE 0 RET-80086A06 12:38:08.6321021408 00.0000049375 =016055=  
00054 QR AP F801 EIBAM ENTRY PUT_CONTAINER RET-8008661A 12:38:08.6321160314 00.0000138906 =016056=  
00054 QR PG 1700 PGCH ENTRY INQUIRE_CHANNEL CHANNEL_NAME(THEMISCHNL) RET-2654A557 12:38:08.6321264377 00.0000104062 =016057=  
00054 QR PG 1701 PGCH EXIT INQUIRE_CHANNEL/EXCEPTION REASON(CHANNEL_NOT_FOUND) CONTAINER_POOL_TOKEN(00000000) RET-2654A557 12:38:08.6321296877 00.0000032500 =016058=  
00054 QR PG 1700 PGCH ENTRY CREATE_CHANNEL CHANNEL_NAME(THEMISCHNL) LINK_LEVEL(CURRENT) CURRENT_CHANNEL(NO) RET-2654A3E9 12:38:08.6321351564 00.0000054687 =016059=  
00054 QR SM 0301 SMGF ENTRY GETMAIN SUBPOOL_TOKEN(3592A41C , 0000006A) GET_LENGTH(40) SUSPEND(YES) INITIAL_IMAGE(00) REMARK (CHCB) LOCK_POOL(YES) RET-ASC876E2 12:38:08.6321437502 00.0000085937 =016060=  
00054 QR SM 0302 SMGF EXIT GETMAIN/OK ADDRESS(26E52030) RET-ASC876E2 12:38:08.6327287814 00.0005850312 =016061=  
00054 QR PG 1800 PGCP ENTRY CREATE_CONTAINER_POOL CCSID(25) IMPORTED(NO) CHANNEL_RELATED(YES) RET-ASC87804 12:38:08.6329509064 00.0002221250 =016062=  
00054 QR SM 0301 SMGF ENTRY GETMAIN SUBPOOL_TOKEN(3592A4E8 , 0000006B) SUSPEND(YES) INITIAL_IMAGE(00) REMARK(CPCB) LOCK_POOL (YES) RET-ASC8A720 12:38:08.6329601564 00.0000092500 =016063=  
00054 QR SM 0302 SMGF EXIT GETMAIN/OK ADDRESS(26E53030) RET-ASC8A720 12:38:08.6329657814 00.0000056250 =016064=  
00054 QR PG 1801 PGCP ENTRY CREATE_CONTAINER_POOL/OK POOL_TOKEN(26E53030) RET-ASC87804 12:38:08.6329700783 00.0000042968 =016065=  
00054 QR PG 1701 PGCH EXIT CREATE_CHANNEL/OK CHANNEL_TOKEN(26E52030) CONTAINER_POOL_TOKEN(26E53030) RET-2654A3E9 12:38:08.6329754377 00.0000053593 =016066=  
00054 QR PG 1900 PGCR ENTRY PUT_CONTAINER_POOL_TOKEN(26E53030) CONTAINER_NAME(THEMISCONT) CALLER(EXEC) DATATYPE(CHAR) ITEM_DATA (27D0F8D0 , 00000020) RET-26545077 12:38:08.6329909377 00.0000155000 =016067=  
00054 QR PG 1900 PGCR EVENT LOCATE_CONTAINER CRCB_PTR(00000000) CONTAINER( ) RET-26545077 12:38:08.6329997971 00.0000088593 =016068=  
00054 QR SM 4201 S2GF ENTRY GETMAIN SUBPOOL_TOKEN(00000048 40804584 , 00000000.0000006F) GET_LENGTH(1000) SUSPEND(YES) REMARK (CSDB) LOCK_POOL(YES) LMLM_ADDRESS(26D53380) RET-25C40937 12:38:08.6330268908 00.0000270937 =016069=
```

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



3/4/2015

21

This slide represents one EXEC CICS command with all possible Level 1 and 2 entries turned on. The intent is to just show the amount of trace data with is possible with the default option settings. Also this illustrates the number of **User Exit (UEH module)** entries which may be produced. There are many IBM and vendor products which implement **XEIN**, **XEIOU**, **XRMIN**, and **XRMOUT** user exits. Can you imagine the number of entries produced by these user exits. This is why we have turned off these entries.

CICS Command Entry with UE Enabled



EXEC CICS command trace with inclusion of user exit trace entries:

```

00054 QR SM 4202 S2GF EXIT GETMAIN/OK ADDRESS(00000048_40D01000) RET-25C40937 12:38:08.6330407814 00.0000138906 =016070=
00054 QK PG 1921 PGCR EVENT PUT_CONTAINER_DATA RET-26545077 12:38:08.6330536096 00.0000128281 =016071=
00054 QR PG 1901 PGCR EXIT PUT_CONTAINER/OK CONTAINER_TOKEN_OUT(26DB20A0) GENERATION_NUMBER(1) INITIAL_GENERATION(1)
00054 QR AP F802 EIBAM EXIT PUT_CONTAINER RESP=0 RESP2=0 RET-26545077 12:38:08.6330575158 00.0000039062 =016072=
00054 QR AP D500 UEH EVENT LINK-TO-USER-EXIT-PROGRAM EQAXEIOU AT EXIT POINT XEIOU RET-80086FLA 12:38:08.6330628439 00.0000053281 =016073=
00054 QR AP D501 UEH EVENT RETURN-FROM-USER-EXIT-PROGRAM EQAXEIOU WITH RETURN CODE 0 RET-80088772 12:38:08.6330871252 00.0000242812 =016074=
00054 QR AP D500 UEH EVENT LINK-TO-USER-EXIT-PROGRAM EQANXOU AT EXIT POINT XEIOU RET-80088772 12:38:08.6330943127 00.0000071875 =016075=
00054 QR AP D501 UEH EVENT RETURN-FROM-USER-EXIT-PROGRAM EQANXOU WITH RETURN CODE 0 RET-80088772 12:38:08.6331005627 00.0000062500 =016076=
00054 QR AP E110 EISR ENTRY TRACE_EXIT PARM_ADDRESS(27D0A7D0) RET-80088772 12:38:08.6331062971 00.0000057343 =016077=
00054 QR AP E161 EXEC EXIT PUT_CONTAINER('THEMISCONT ' AT X'27D0FA20') CHANNEL('THEMISCHNL ' AT X'27D0FA10') FROM( AT X'27D0F8D0') FLENGTH(32 AT X'27D0FA30') DATATYPE(1019 AT X'A7D0A960') RESP(0) RESP2(0) COBOL11 SINT_#(00049) RET-8008889C 12:38:08.6331598439 00.0000428906 =016079=
00054 QR AP E111 EISR EXIT TRACE_EXIT/OK RET-8008889C 12:38:08.6331635314 00.0000036875 =016080=
00054 QR AP 00E1 EIP EXIT PUT-CONTAINER OK REQ(00FA) FIELD-A(00000000 ...) FIELD-B(00003416 ...) RET-A7419026 12:38:08.6331678283 00.0000042968 =016081=
00054 QR BOUNDARY(0200)

```


Complete your session evaluations online at www.SHARE.org/Seattle-Eval



3/4/2015

22

This slide is a continuation of the EXEC CICS command with all possible trace entries. Again, notice the number of user exit trace entries which are produced by the common utilized user exit points. Also, please notice the last three trace entries, **AP E161**, **AP E111**, and **AP 00E1**. They correspond to the first three trace entries on the previous slide, **AP 00E1**, **AP E110**, and **AP E160**. These are produced by STNTREI=(1,2) and SPCTREI=(1,2) SIT overrides and provide the most information about the commands executed by the application programs.



Tokens in Trace Entries

Tokens:

```

XM QR DS 0012 DSKE ENTRY TASK_REPLY TASK_TOKEN(26C6A800) ATTACH_TOKEN(3596A080)
RET-A5504230 08:36:29.8023245156 00.0000403281 =000778=

XM QR XM 1305 XMTA ENTRY TASK_REPLY USER_TOKEN(26DFA900) TASK_TOKEN(05800007) SUSPEND_TOKEN(05800007)
RET-A552060C 08:36:29.8023403750 00.0000158593 =000779=

XM QR SM 0301 SMGF ENTRY GETMAIN SUBPOOL_TOKEN(258A88E4 , 00000111) GET_LENGTH(F5C) SUSPEND(YES) INITIAL_IMAGE(00)
RET-ASRE3768 08:36:29.8022307968 00.0003813468 =000781=
XM QR SM 1201 SMPQ ENTRY ALLOCATE_PAGEPOOL_STORAGE SUBPOOL_TOKEN(258A88E4) GET_LENGTH(F60) SUSPEND(YES)
RET-2553787F 08:36:29.8027561406 00.0000253437 =000782=


00053 QR DS 0004 DSSR ENTRY SUSPEND SUSPEND_TOKEN(05800009) RESOURCE_TYPE(ZCLOWAIT) PURGEABLE(YES) DEADLOCK_ACTION(INHIBIT)
WLM_WAIT_TYPE(IDLE) RESOURCE_NAME(OPHZARQ1)
RET-A613EC54 08:36:46.0700694064 00.0000132656 =003009=

TCP QR DS 0004 DSSR ENTRY RESUME SUSPEND_TOKEN(05800009) RET-A61C5560 08:36:49.053582968 00.0000156093 =003066=
TCP QR DS 0005 DSSR EXIT RESUME/OK RET-A61C5560 08:36:49.0535753437 00.0000170468 =003067=

00053 QR DS 0005 DSSR EXIT SUSPEND/OK RET-A613EC54 08:36:49.0536541250 00.0000266875 =003074=

```

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



3/4/2015 23

Tokens which may be of use:


A **Token** is an identifier for an object. This enables reference to objects by different domains in a manner unique to each domain. **Tokens may be used to analyze task interaction from the trace evidence.** **Objects** that have tokens associated with them include:

- Domains** A unique number from 1 to 21, KE Table Entry for the domain. Not the same as Domain ID, but there is a one to one correspondence
- Locks** Returned by **ADD_LOCK** function of the **LMLM** gate in LM domain. Address of Lock Management Block used to lock a logical resource (**SHARED** or **EXCLUSIVE**)
- Storage** Returned by **ADD_SUBPOOL** function of the **SMAD** gate in SM domain. Address of Subpool Control Area (**SCA**) plus a unique number for **Subpools** each pool created.
- Programs** Returned by **DEFINE_PROGRAM** & (optionally) the **ACQUIRE_PROGRAM** function of the **LDLD** gate in LD domain. Address of Current Program Entry block (**CPE**) used to manage defined programs

There are many tokens associated with a task. Consider the token from the perspective of the owning domain and their part within task creation as follows:

Domain Involved/

<u>Label In Trace</u>	<u>Description</u>
Dispatcher (DS) ATTACH_TOKEN	For CREATE_TASK : address of the DTA . The value is returned on the TCB_REPLY and TASK_REPLY functions of the KEDS format gate in DS (in module DSKE). It is identified as DS_TASK in the Kernel Domain Summary.
TASK_TOKEN	The value found on the DSKE ENTRY gate for function TASK_REPLY contains the address of a KE TTE (task table entry or " <i>slot</i> "). The value returned of the DSAT EXIT gate for function ATTACH contains the <i>block</i> , <i>cell</i> and <i>count</i> found on the DS Summary as the DS_TOKEN column.
TRANSACTION_TOKEN	The value found on the DSAT ENTRY (Version 4.x only) gate for function ATTACH contains the address of the TXN (Transaction) in the first full word and the packed task number in the second.



Tokens in Trace Entries

Tokens:

```

XM  QR  DS 0012 DSKE  ENTRY TASK_REPLY TASK_TOKEN(26C6A800) ATTACH_TOKEN(3596A080)
                                           RET-A5504230 08:36:29.8023245156 00.0000403281 =000778=

XM  QR  XM 1305 XMTA  ENTRY TASK_REPLY USER_TOKEN(26DFA900) TASK_TOKEN(05800007) SUSPEND_TOKEN(05800007)
                                           RET-A552060C 08:36:29.8023403750 00.0000158593 =000779=


XM  QR  SM 0301 SMGF  ENTRY GETMAIN SUBPOOL_TOKEN(258A88E4 , 00000111) GET_LENGTH(F5C) SUSPEND(YES) INITIAL_IMAGE(00)
                                           RET-ASBE3768 08:36:29.8027307968 00.0003813468 =000781=
XM  QR  SM 1201 SMPQ  ENTRY ALLOCATE_PAGEPOOL_STORAGE SUBPOOL_TOKEN(258A88E4) GET_LENGTH(F60) SUSPEND(YES)
                                           RET-25537B7F 08:36:29.8027561406 00.0000253437 =000782=

00053 QR  DS 0004 DSSR  ENTRY SUSPEND SUSPEND_TOKEN(05800009) RESOURCE_TYPE(ZCLOWAIT) PURGEABLE(YES) DEADLOCK_ACTION(INHIBIT)
                                           WLM_WAIT_TYPE(IDLE) RESOURCE_NAME(OPHZARQ1)
                                           RET-A613EC54 08:36:46.0700694064 00.0000132656 =003009=

TCP  QR  DS 0004 DSSR  ENTRY RESUME SUSPEND_TOKEN(05800009) RET-A61C5560 08:36:49.0535582968 00.0000156093 =003066=
TCP  QR  DS 0005 DSSR  EXIT RESUME/OK RET-A61C5560 08:36:49.0535753437 00.0000170468 =003067=

00053 QR  DS 0005 DSSR  EXIT SUSPEND/OK RET-A613EC54 08:36:49.0536541250 00.0000266875 =003074=

```



Complete your session evaluations online at www.SHARE.org/Seattle-Eval

3/4/2015 24

Tokens which may be of use (continued):

**Domain Involved/
Label In Trace**

Description

**Storage Manager (SM)
SUBPOOL_TOKEN**

The value found on the **SMGF ENTRY** gate for function **GETMAIN** contains the address of the **SCA** (Subpool Control Area) in the first full word and the Subpool identifier in the second.

**Transaction Manager
USER_TOKEN**

The value returned on the **XMTA ENTRY** gate for function **TASK_REPLY** contains the address of the **TXN**. The **(XM)** value returned on the **DSAT ENTRY** gate for function **ATTACH** contains the address of the **TXN**.

SUSPEND_TOKEN

The value returned on the **XMTA ENTRY** gate for function **TASK_REPLY** contains the *block, cell and count* used by **DSSR** functions **SUSPEND** and **RESUME**.

TASK_TOKEN

The value returned on the **XMTA ENTRY** gate for function **TASK_REPLY** contains the *block, cell and count* found on the DS Summary as the **DS_TOKEN** column.

- Trace Entries Representing Problems

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

3/4/2015

Trace Entries Representing Problems



Exception Trace Entries:

User generated entry:

```
AP 004D USER *EXCU - APPLICATION-PROGRAM-EXCEPTION - ABIGRESS - .....
TASK-00054 KE_NUM-006B TCB-QR /008E6968 RET-80086F9E TIME-12:13:41.2270181096 INTERVAL-00.0000172968 =041478=
1-0000 E4E2C5D9 C5E7C3 ..... *USEREXC *
2-0000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 ..... *
0020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 ..... *
..... *
05C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 ..... *
05E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 ..... *
3-0000 C1C2C9C7 D4C5E2E2 ..... *ABIGRESS *
```

(1) Data areas are at user discretion

CICS generated entry:

```
SM 0011 SMMF *EXC* - storage_check_failed_on_freemain_request - FUNCTION(FREEMAIN) ADDRESS(28011008) CALLER(EXEC) EXEC_KEY(USER)
TASK-00054 KE_NUM-006B TCB-QR /008E6968 RET-266C7451 TIME-12:13:41.2278113283 INTERVAL-00.0000202343 =041487=
1-0000 00980000 00000011 00000000 00000000 84090000 00000000 040001AA 00000009 ..... *Q.....XQ.WS.....
0020 25AA0560 28011008 00000004 25AA0100 2800A7D8 26E64A01 A66A1130 2800FA42 ..... *.....XQ.WS.....
0040 2800FA42 00000003 2800A7E8 266A5140 26E64780 266A595A 26E64AD4 28000750 ..... *.....XY...W...I.WSM...&
0060 00000004 25AA0560 266A7DA6 00000002 25AA0100 2800A7E8 26E64A64 26E64AD4 ..... *.....W...XY.WS...WSM*
0080 2800A7D8 26E64A64 26E64AD4 2800A7D8 26E64A90 00000000 ..... *XQ.WS.WSM.XQ.WS.....
2-0000 28011000 ..... *
3-0000 40C3D6E4 D987C540 A07E7E5C 00000000 0000E767 40404040 40400000 00000000 ..... *COURSE ==*.....SP
0020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 ..... *
0040 00000000 00000000 00000000 00000000 00000000 00000000 00000000 ..... *
0060 00000000 00000000 00000000 00000000 00000000 00000000 00000000 ..... *
4-0000 40E3C8C5 D4C9E240 C3C9C3E2 40E3E2F5 ..... *THEMIS CICS TS5 *
```

(2) Data areas are:
1 - Parameter list
2 - Address of storage area
3 - First 64 bytes of area
4 - Last 16 bytes of area

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



There are literally hundreds if not thousands of exception trace entries possible within CICS. Most of which indicate a hard problem like an abend or critical resource shortage. However, many represent some lesser performance issue like thread or string wait conditions.

The application itself can generate exception trace entries, like the first one shown on this slide. Instead of the trace type being *EXC*, it will be *EXCU to indicate an application generated entry. The trace data, resource information, and interpretation information are dependent on what the application program has supplied.

The second entry on this slide, trace type *EXC*, represents a storage related problem. This entry is generated by CICS itself when an application program executes a FREEMAIN command and the storage area is damaged. Storage violations are only checked when the storage areas for a task are being freed. At the front and end of each acquired piece of storage is an eight byte field known as the **Check Zone**. These areas normally contain the storage subpool identifier, a literal character value. If either check zone does not contain this value then CICS considers the storage area to be violated. There is no way for CICS to know the extent of the storage overlay, just the fact that the first and last eight bytes are invalid.

Also, in looking at the "storage_check_failed.." entry, you will notice several associated data areas. These areas are somewhat documented in the CICS TS Trace Entries manual. While not all areas are thoroughly covered (the parameter list as an example), the key ones will have some meaningful description. In this entry, data areas 2, 3 and 4 are significant, respectively providing the address, first 64 bytes and last 16 bytes of the data area. As you can see, both the leading and trailing check zones do not contain the proper values.

Trace Entries Representing Problems

Exception Trace Entries:

CICS generated entry:

SM 0FDC SMAR *EXC* - Storage_check_failed_at_address - 28011210 FUNCTION(RELEASE_TRANSACTION_STG)

```
TASK-XM KE_NUM-0068 TCB-QR /008E6968 RET-A56599AC TIME-12:13:55.2103549375 INTERVAL-00.0000163750 =049887=
1-0000 00280000 00000001 00000000 00000000 00000000 00000000 02000100 00000000 *.....1.....*
0020 00000000 00000000 *.....*
2-0000 28011210 *.....*
3-0000 00000210 *.....*
4-0000 40C3D6E4 09E2C540 407E7E5C 00000000 00000000 00000000 00000000 * COURSE =m*.....*
0020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
.....*
00E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0100 00000000 00000000 *.....*
5-0000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
.....*
00E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0100 E4F0F0F0 F0F0F0F0 *U0000054.....*
6-0000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
.....*
03C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
03E0 00000000 00000000 00000000 5C7E7E40 40E3C8C5 D4C9E240 C3C9C3E2 40E3E2F5 *.....* =m THEMIS CICS T55*
7-0000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
.....*
(1) Data areas are:
.....*
0300 00000000 0 1 - Parameter list *.....*
0320 00000000 0 2 - Address of storage area *.....*
3 - Length of storage area
4 - First 512 (max.) bytes of area
5 - Last 512 (max.) bytes of area
6 - Data preceding area (max. 1K)
7 - Data following area (max. 1K)
```

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



27

This slide contains the exception trace entry produced by CICS for a storage violation when the storage is being freed at task end. This is normally when the majority of storage violations occur since most application programs do not issue FREEMAIN commands. As with the previous storage check failed trace entry, the data areas here may be useful. Data area 1 is the address of the storage area, while data area 2 is its length (in hex). Data areas 4 and 5 show up to 512 bytes at the start and end of the storage area, respectively. Also shown in data areas 6 and 7 area the areas before and after the violated area, up to 1024 bytes each. The latter two areas are presented in case the data may assist with finding the offending application program.

Trace Entries Representing Problems



Exception Trace Entries:

```

CICS generated entry:
00053 QR  AP 1942 APLI *EXC* Program-Check  FUNCTION(START_PROGRAM) PROGRAM(THMPGMC) CEDF_STATUS(CEDF) EXECUTION_SET(FULLAPI)
ENVIRONMENT_TYPE(EXEC) SYNCONRETURN(NO) LANGUAGE_BLOCK(26FFE66C) COMMAREA(00000000 ,
00000000) LINK_LEVEL(2) SYSEIB_REQUEST(NO) RET-A501DE58 10:45:58.0036145002 00.0430751408#=000751=

AP 1942 APLI *EXC* - Program-Check  FUNCTION(START_PROGRAM) PROGRAM(THMPGMC) CEDF_STATUS(CEDF) EXECUTION_SET(FULLAPI)
ENVIRONMENT_TYPE(EXEC) SYNCONRETURN(NO) LANGUAGE_BLOCK(26FFE66C) COMMAREA(00000000 , 00000000) LINK_LEVEL(2)
SYSEIB_REQUEST(NO)

TASK-00053 KE_NUM-0068 TC8-QR /008E6968 RET-A501DE58 TIME-10:45:58.0036145002 INTERVAL-00.0430751408# =000751=
1-0000 01880000 000000DA 00000000 00000000 881B4750 00000000 02000100 E3C8D4D7 *.....&.....THMP*
0020 C704C340 00000000 00000000 29223030 00000001 01510202 2751C000 26FFE66C *GMC .....{...W*
.....
0180 00000000 0C900000 00000000 00003000 00000000 00000038 C0000000 26E5E148 *.....{...V..*
0100 00000000 26E500D8 00000000 25838E80 00000000 25844300 *.....V.Q...C.....d..*
2-0000 F0C3F761 C102C5C1 018400C7 00000000 C4C6C8C1 D703C9F1 00000000 26200000 *0C7/AKEA.d.g....DFMAPL11.....*
0020 00000000 25A8E100 00000000 26E4E800 35A1B080 00000001 00000000 FFFFFFFF *...y.....UY.....*
0040 7952000 80000000 00000000 275269EE 00060007 00000000 00000000 00000000 *..n.....*
.....
0280 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0240 41FB0068 00000001 00030003 C0000000 00000000 27E1748E 25843200 0000053C *.....{.....d.....*
02C0 00000000 7F4E1800 00000000 00000000 *...*.....*
    
```

(1) Data areas are:
 1 - Parameter list
 2 - KE Error Data Area

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



This slide contains the exception trace entry produced by CICS for a storage violation when the storage is being freed at task end. This is normally when the majority of storage violations occur since most application programs do not issue FREEMAIN commands. As with the previous storage check failed trace entry, the data areas here may be useful. Data area 1 is the address of the storage area, while data area 2 is its length (in hex). Data areas 4 and 5 show up to 512 bytes at the start and end of the storage area, respectively. Also shown in data areas 6 and 7 area the areas before and after the violated area, up to 1024 bytes each. The latter two areas are presented in case the data may assist with finding the offending application program.

Trace Entries Representing Problems



Exception Trace Entries:

CICS generated entry:
00053 QR AP 0790 SRP *EXC* PROGRAM_CHECK RET-A5A8170C 10:45:58.0036676877 00.0000531875 =000752=

AP 0790 SRP *EXC* - PROGRAM_CHECK

```
TASK-00053 KE_NUM-006B TCB-QR /008E6968 RET-A5A8170C TIME-10:45:58.0036676877 INTERVAL-00.0000531875 =000752=  
1-0000 F0C3F761 C1D2C5C1 018400C7 00000000 C4C6C8C1 D7D3C9F1 00000000 26200000 *OC7/AKEA,d.G...DFHAPLI1.....*  
0020 00000000 25A8E100 00000000 26E4E800 35A1B080 00000001 00000000 FFFFFFFF *.....y.....UY.....*  
0040 07952000 80000000 00000000 275269EE 00060007 00000000 00000000 *.....n.....*  
.....*  
0280 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*  
02A0 41FB8968 00000001 00030003 C0000000 00000000 27E1748E 25843200 0000053C *.....{.....d.....*  
02C0 00000000 7F4E1800 00000000 00000000 *....."+.....*
```

(1) Data areas are:
1 - KE Error Data Area

00053 QR AP 0781 SRP *EXC* ABEND_ASRA PROGRAM(THMPGMC) OFFSET(0000123E) EXEC_KEY(USER) RET-A5A8170C 10:45:58.0043387502 00.0000138437 =000759=

AP 0781 SRP *EXC* - ABEND_ASRA PROGRAM(THMPGMC) OFFSET(0000123E) EXEC_KEY(USER)

```
TASK-00053 KE_NUM-006B TCB-QR /008E6968 RET-A5A8170C TIME-10:45:58.0043387502 INTERVAL-00.0000138437 =000759=  
1-0000 E3C804D7 C7D4C340 F0C3F761 C1D2C5C1 0000123E 00020781 *THMPGMC OC7/AKEA.....a *  
2-0000 00 *.....*  
3-0000 F0C3F761 C1D2C5C1 018400C7 00000000 C4C6C8C1 D7D3C9F1 00000000 26200000 *OC7/AKEA,d.G...DFHAPLI1.....*  
0020 00000000 25A8E100 00000000 26E4E800 35A1B080 00000001 00000000 FFFFFFFF *.....y.....UY.....*  
0040 07952000 80000000 00000000 275269EE 00060007 00000000 00000000 *.....n.....*  
.....*  
0280 00000000 (1) Data areas are: 0000 *.....*  
02A0 41FB8968 1 - Program, KE error code, offset 053C *.....{.....d.....*  
02C0 00000000 2 - Storage hit indicator *....."+.....*  
3 - KE Error Data Area
```

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



This slide contains the exception trace entry produced by CICS for a storage violation when the storage is being freed at task end. This is normally when the majority of storage violations occur since most application programs do not issue FREEMAIN commands. As with the previous storage check failed trace entry, the data areas here may be useful. Data area 1 is the address of the storage area, while data area 2 is its length (in hex). Data areas 4 and 5 show up to 512 bytes at the start and end of the storage area, respectively. Also shown in data areas 6 and 7 area the areas before and after the violated area, up to 1024 bytes each. The latter two areas are presented in case the data may assist with finding the offending application program.


- Application Profiling

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

3/4/2015

- The next few slides represent our transaction THMY, which interacts with one of our DB2 regions.
- One of the goals here is to illustrate how you can tell just where your application program is in the scheme of things.
- Another goal is to provide insight into determining if the potential issue is part of CICS or some other resource manager.

Transaction Profile (THMY) with Trace Entries



SHARE
in Seattle 2015

Start of the task:

```

XM QR  DS 0012 DSKE ENTRY TASK_REPLY TASK_TOKEN(26046800) ATTACH_TOKEN(35A75980) RET-A5604230 11:16:30.2111084531 00.0000841093 =001818=
XM QR  XM 1305 XMTA ENTRY TASK_REPLY USER_TOKEN(2744DD00) TASK_TOKEN(058C0001) SUSPEND_TOKEN(058C0001) RET-A562073C 11:16:30.2111216093 00.0000131562 =001819=

XM QR  PG 0801 PGXM ENTRY INITIALIZE_TRANSACTION RET-A5657D36 11:16:30.2129788437 00.0000137968 =001824=
XM QR  PG 0802 PGXM EXIT INITIALIZE_TRANSACTION/OK RET-A5657D36 11:16:30.2129859375 00.0000070937 =001825=
XM QR  SM 0301 SMGF ENTRY GETMAIN GET_LENGTH(650) SUSPEND(YES) INITIAL_IMAGE(00) REMARK(AP_USER) STORAGE_CLASS(TASK) RET-A62A3786 11:16:30.2130038906 00.0000179531 =001826=
XM QR  SM 0302 SMGF EXIT GETMAIN/OK ADDRESS(2940000R) RET-A62A3786 11:16:30.2142765312 00.0012726406 =001827=
00056 QR SM 0301 SMGF ENTRY GETMAIN GET_LENGTH(FA0) SUSPEND(YES) REMARK(EI_TRACE) STORAGE_CLASS(USER) RET-A62A3A24 11:16:30.2143085781 00.0000320468 =001828=
00056 QR SM 0302 SMGF EXIT GETMAIN/OK ADDRESS(29400668) RET-A62A3A24 11:16:30.2143626875 00.0000541093 =001829=


Start of uow plus initial program load:
00056 QR RM 0209 RMUC EVENT Remote_UOW_id_created UOW_ID(150CE2F0E6F148E3C3C9C3E2C1F36280B87F95E800010000000000) RET-A565812E 11:16:30.2144970468 00.0000104687 =001835=

00056 QR AP 2520 ERM ENTRY CALL-TRUES-FOR-TASK-START RET-A62A3B88 11:16:30.2157517656 00.0000353125 =001855=
00056 QR AP 2521 ERM EXIT CALL-TRUES-FOR-TASK-START RET-A62A3B88 11:16:30.2168172187 00.0000284843 =001870=

00056 QR PG 0901 PGPB ENTRY INITIAL_LINK PROGRAM_NAME(THMPGMB) AUTOINST_INITIAL(YES) RET-A5658DAE 11:16:30.2168324843 00.0000065781 =001872=
00056 QR DS 0004 DSSR ENTRY SUSPEND SUSPEND_TOKEN(058C0001) RESOURCE_TYPE(PROGRAM) PURGEABLE(YES) RESOURCE_NAME(THMPGMB) RET-ASD34724 11:16:30.2168656718 00.0000331875 =001873=
Load time: .0212 secs
00056 QR DS 0005 DSSR EXIT SUSPEND/OK RET-ASD34724 11:16:30.2381294687 00.0000951875 =002140=
00056 QR LD 0001 LDLD ENTRY ACQUIRE_PROGRAM PROGRAM_TOKEN(00000048_40CC8FB8) RET-ASD35A9E 11:16:30.2381559218 00.0000264531 =002141=
00056 QR LD 0002 LDLD EXIT ACQUIRE_PROGRAM/OK ENTRY_POINT(A7622000) LOAD_POINT(27622000) PROGRAM_LENGTH(9808) PROGRAM_ATTRIBUTE(REUSABLE) LOCATION(ESDSA) COPY_STATUS(OLD_COPY) FETCH_TIME() LIBRARY (DFHRPL) BUNDLE_INSTALLED_LIB(NO) PRIVATE_LIBRARY(NO) RET-ASD35A9E 11:16:30.2381735156 00.0000175937 =002142=

```

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



SHARE
in Seattle 2015

32


Now to do a little application profiling with trace data. Our first example is a transaction which performs DB2 requests. While we will not go through the complete sequence of events, we will cover several key items in the CICS / DB2 interface. We will pay close attention to the bolded, underlined portions of these trace entries.

First let's call your attention to the first two trace entries, "**DS 0012 DSKE ENTRY TASK_REPLY**" and "**XM1305 XMTA ENTRY TASK_REPLY**". All transaction in CICS will start with these two trace entries. They represent the task passing any DS domain constraints like max. task, plus XM domain constraints like TRANCLASS. Although we will not show then, you will see the two corresponding "**EXIT TASK_REPLY**" trace entries as the last two trace entries for the transaction.

The next two high-lighted entries, **PG 0801** and **PG 0802**, represent the start of the transaction from a monitor, statistical point of view. At this point CICS will begin to capture resource utilization information for the task. Next you will notice that a **GETMAIN** is issued for **REMARK(AP_USER)** type of storage. This is where the AP domain begins to recognize the task and assign a task number, 56 in this case. You see another GETMAIN being done with a type of **REMARK(EI_TRACE)**. It should be noted that this is CICS performing the command since the initial application program is yet to be invoked. However, two key attributes on any GETMAIN trace entries will be the **GET_LENGTH()** and **ADDRESS()**. These attributes proved the length of the acquired storage and its address, respectively. When working with a storage violation dump, dump codes **SM0102** and **SM0103**, these can be used to help find out which transaction really owns the storage area being violated.

We have left out a few entries as indicated by the sequence numbers =001829= and =001835=. The trace point **RM 0209** provides us with the assigned, unique **UOW_ID()** or **Unit of Work Identifier**. This can be used to assist with resolution of indoubt or shunted UOWs. The UOW_ID is also used as a token to manage UOWs across related subsystems like DB2 or WebSphere MQ.

Transaction Profile (THMY) with Trace Entries



SHARE
in Seattle 2015

Start of the task:

```

XM QR DS 0012 DSKE ENTRY TASK_REPLY TASK_TOKEN(26046800) ATTACH_TOKEN(35A75980) RET-A5604230 11:16:30.2111084531 00.0000841093 =001818=
XM QR XM 1305 XMTA ENTRY TASK_REPLY USER_TOKEN(2744DD00) TASK_TOKEN(058C0001) SUSPEND_TOKEN(058C0001) RET-A562073C 11:16:30.2111216093 00.0000131562 =001819=

XM QR PG 0801 PGXM ENTRY INITIALIZE_TRANSACTION RET-A5657D36 11:16:30.2129788437 00.0000137968 =001824=
XM QR PG 0802 PGXM EXIT INITIALIZE_TRANSACTION/OK RET-A5657D36 11:16:30.2129859375 00.0000070937 =001825=
XM QR SM 0301 SMGF ENTRY GETMAIN GET_LENGTH(650) SUSPEND(YES) INITIAL_IMAGE(00) REMARK(AP_USER) STORAGE_CLASS(TASK) RET-A62A3786 11:16:30.2130038906 00.0000179531 =001826=
XM QR SM 0302 SMGF EXIT GETMAIN/OK ADDRESS(2940000R) RET-A62A3786 11:16:30.2142765312 00.0012776406 =001827=
00056 QR SM 0301 SMGF ENTRY GETMAIN GET_LENGTH(FA0) SUSPEND(YES) REMARK(EI_TRACE) STORAGE_CLASS(USER) RET-A62A3A24 11:16:30.2143085781 00.0000320468 =001828=
00056 QR SM 0302 SMGF EXIT GETMAIN/OK ADDRESS(29400668) RET-A62A3A24 11:16:30.2143626875 00.0000541093 =001829=


Start of uow plus initial program load:
00056 QR RM 0209 RMUC EVENT Remote_UOW_id_created UOW_ID(150CE2F0E6F148E3C3C9C3E2C1F36280B87F95E800010000000000) RET-A565812E 11:16:30.2144970468 00.0000104687 =001835=

00056 QR AP 2520 ERM ENTRY CALL-TRUES-FOR-TASK-START RET-A62A3B88 11:16:30.2157517656 00.0000353125 =001855=
00056 QR AP 2521 ERM EXIT CALL-TRUES-FOR-TASK-START RET-A62A3B88 11:16:30.2168172187 00.0000284843 =001870=

00056 QR PG 0901 PGPB ENTRY INITIAL_LINK PROGRAM_NAME(THMPGMB) AUTOINST_INITIAL(YES) RET-A5658DAE 11:16:30.2168324843 00.0000065781 =001872=
00056 QR DS 0004 DSSR ENTRY SUSPEND SUSPEND_TOKEN(058C0001) RESOURCE_TYPE(PROGRAM) PURGEABLE(YES) RESOURCE_NAME(THMPGMB) RET-A5D34724 11:16:30.2168656718 00.0000331875 =001873=
Load time: .0212 secs
00056 QR DS 0005 DSSR EXIT SUSPEND/OK RET-A5D34724 11:16:30.2381294687 00.0000951875 =002140=
00056 QR LD 0001 LDLD ENTRY ACQUIRE_PROGRAM PROGRAM_TOKEN(00000048_40CC8FB8) RET-A5D35A9E 11:16:30.2381559218 00.0000264531 =002141=
00056 QR LD 0002 LDLD EXIT ACQUIRE_PROGRAM/OK ENTRY_POINT(A7622000) LOAD_POINT(27622000) PROGRAM_LENGTH(9808) PROGRAM_ATTRIBUTE(REUSABLE) LOCATION(ESDSA) COPY_STATUS(OLD_COPY) FETCH_TIME() LIBRARY (DFHRPL) BUNDLE_INSTALLED_LIB(NO) PRIVATE_LIBRARY(NO) RET-A5D35A9E 11:16:30.2381735156 00.0000175937 =002142=

```

Complete your session evaluations online at www.SHARE.org/Seattle-Eval




SHARE
in Seattle 2015

33

The next entry, “**PG 0901 PGPB ENTRY INITIAL_LINK**” indicates that CICS is about to pass control to the first application program associated with the transaction. Notice that the task ends up being suspended as indicated by the “**DSSR ENTRY SUSPEND**” trace entry. The **SUSPEND_TOKEN()** is used as a token for DS domain to keep track of the tasks being put into various wait states, plus to identify which task is being resumed when the wait condition is resolved. The main reason for the task wait state is because the program has to be loaded into memory. By subtracting the time on the DSSR ENTRY trace from the DSSR EXIT trace you can obtain the approximate time it took to complete the load request, **.0212 seconds** in this case. At times one of the mysteries of poor task performance may in fact be the loading of a large application program, i.e. something greater than 1 MB. Loading of large programs will require many I/Os to read in the program from the load library, plus the potential need to perform storage compression or acquire more extents of storage to hold the program.

Finally we have high-lighted the **ENTRY_POINT()** and **LOAD_POINT()** attributes on the “**LD 0002 EXIT**” trace entry. When working with application profiling, these can be quite useful to track back the commands shown in trace back to the actual source program. They can also be used to assist with problem resolution when working with exception trace entries which are the result of an abend.

Transaction Profile (THMY) with Trace Entries



Acquisition of initial storage areas:

```

00056 QR SM 0301 SMGF ENTRY GETMAIN GET_LENGTH(3ACC) SUSPEND(YES) INITIAL_IMAGE(00) REMARK(LE_TWA) STORAGE_CLASS(TASK31)
RET-A6203406 11:16:30.2381156562 00.0001802500 =002147=
00056 QR SM 0302 SMGF EXIT GETMAIN/OK ADDRESS(294018C8)
RET-A6203406 11:16:30.238124687 00.0003268125 =002148=
00056 QR SM 0301 SMGF ENTRY GETMAIN GET_LENGTH(8760) SUSPEND(YES) REMARK(LE_RUNA) STORAGE_CLASS(TASK31)
RET-A6203406 11:16:30.2381156562 00.0000529687 =002149=
00056 QR SM 0302 SMGF EXIT GETMAIN/OK ADDRESS(294053A8)
RET-A6203406 11:16:30.2381156562 00.0000932812 =002150=
00056 QR SM 0C01 SMGF ENTRY GETMAIN GET_LENGTH(5440) SUSPEND(YES) STORAGE_CLASS(USER) CALLER(EXEC)
RET-266CAAA3 11:16:30.2394971111 00.0000188906 =002154=
00056 QR SM 0C02 SMGF EXIT GETMAIN/OK ADDRESS(29400818)
RET-266CAAA3 11:16:30.2396981111 00.0002005000 =002155=

```

Dynamic Storage Area

LE Run Unit Work Area

LE Task Work Area

Some of the commands or DB2 requests performed:

```

00056 QR AP 00E1 EIP ENTRY GET-CONTAINER
BOUNDARY(0200) REQ(0004) FIELD-A(2940A658 . w.) FIELD-B(09003414 . . . .)
RET-A762773A 11:16:30.2445081250 00.0002111562 =002224=
00056 QR AP E160 EXEC ENTRY GET CONTAINER('THEMISCONT ' AT X'2940F9FB') CHANNEL('THEMESCHNL ' AT X'A940F9EB') INTO(
AT X'29410264') FLENGTH(32 AT X'2940FA0B') COBOLII STMT_(00369)
RET-8008661E 11:16:30.2445580000 00.000408760 =002225=
00056 QR AP E161 EXEC EXIT GET CONTAINER('THEMISCONT ' AT X'2940F9FB') CHANNEL('THEMESCHNL ' AT X'A940F9EB')
RESP(0) RESP2(0) COBOLII STMT_(00369)
RET-8008889C 11:16:30.247721734 00.000733281 =002232=
00056 QR AP 00E1 EIP EXIT GET-CONTAINER OK
BOUNDARY(0200) REQ(00F4) FIELD-A(00000000 . . . .) FIELD-B(00003414 . . . .)
RET-A762773A 11:16:30.2477294377 00.0000077031 =002233=
00056 QR AP 2520 ERM ENTRY COBOL-APPLICATION-CALL-TO-TRUE(COSNCSQL . . . .)
RET-A7629260 11:16:30.2477671408 00.0000377031 =002234=
00056 QR DS 0008 DSIT ENTRY ADD_TCB_IDENTITY(L8 -OPEN) MODENAME(L8) CHANGE_MODE_PLIST(26051E20)
RET-A561CFB2 11:16:30.2517097814 00.0000437033 =002294=
00056 QR DS 0009 DSIT EXIT ADD_TCB/OK TCB_TOKEN(2595A400 , 000000001)
RET-A561CFB2 11:16:30.2518279221 00.0001020002 =002296=
DSTCB L8004_DS_0012_DSKE ENTRY TCB_REPLY TASK_TOKEN(27455100) ATTACH_TOKEN(2595A400)
RET-A567351C 11:16:30.2517097814 00.0000437033 =002294=
00056 L8004 AP 3180 D2EX1 ENTRY APPLICATION REQUEST - EXEC SQL OPEN
RET-800960D6 11:16:30.2518279221 00.0001020002 =002296=


```

Get Time: .0032 secs.

DB2 CALL initiated

Actual DB2 CALL

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



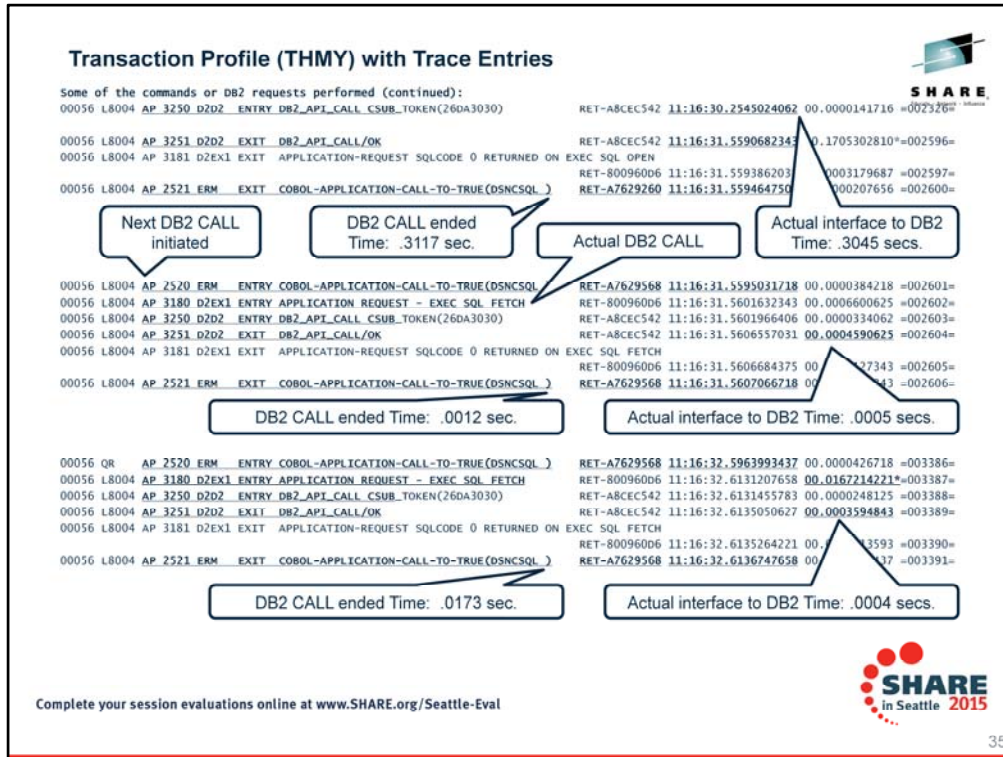
The first four high-lighted trace entries on this slide show two initial GETMAINS performed at the start of a task for LE related areas. While these could be important when using LE data areas to debug applications, our main purpose is to point out the **GET_LENGTH()** and **ADDRESS()** attributes. When working with a storage violation dump, it may be necessary to find the original GETMAIN request for a storage area to determine which task owns the storage.

The third GETMAIN request is for the copy of the “dynamic storage area” (COBOL working storage) for the initial program. This storage area is described in the compile listing for the application program involved and can be useful when you need to determine values for attributes used within the program. However, that is the subject for another presentation or our CICS Problem Resolution course (sorry for the little plug).

The first actual command our application issues is a **GET CONTAINER** command. Here you see the EIP ENTRY and the EIP EXIT trace entries. By subtracting the time on the ENTRY from the time on the EXIT you can tell the amount of time it takes to process the command. While this may seem to be minimal, so was the data area involved. All containers are kept “above the bar” until retrieved and then **moved to below the bar** for use. Since containers can be very large (100s of megabytes), this move operation can be quite significant.

The next high-lighted entry, **ERM ENTRY**, represents the start of a DB2 call. Besides noting the **RET-** attribute, make a mental note of the **TIME** value. We will use the time value soon.

The last high-lighted entry, **D2EX1 ENTRY**, represents the actual DB2 call being performed, an **SQL OPEN** request in this case. This may be considered to be start of the DB2 request, but we are still within the resource manager code in the CICS region.



The first entry, **D2D2 ENTRY**, on this slide is truly the start of the DB2 request. It is produced just prior to transfer to the DB2 region itself. The next entry, **D2D2 EXIT**, is produced upon return from the DB2 region. By subtracting the ENTRY time from the EXIT time you can determine how long the request took to process in the DB2 region. In our example, over .3 seconds. This is a reasonable example of how the coding of the SQL request can have a profound affect on response times. Our SQL OPEN has an ORDER BY clause which forces a sort of approximately 50,000 rows.

Next you see the **ERM EXIT** trace entry. This signifies the actual return back to the application program. One could also consider this to be the total DB2 response time since the resource manager is in fact part of the DB2 interface. We will leave this up to you to determine. As you can see, the actual difference in time is minimal.

You then see the next DB2 call being performed, an **SQL FETCH** call. Again, the actual DB2 time is only .0005 seconds, D2D2 ENTRY to D2D2 EXIT trace entries. However if you consider using the ERM ENTRY to ERM EXIT trace entries the time is over double.

On the same slide we have included a third DB2 request to simply show the fact that there may be small variances in each DB2 request.

Transaction Profile (THMY) with Trace Entries



Some of the commands or DB2 requests performed (continued):

```
00056 QR AP 2520 ERM ENTRY COBOL-APPLICATION-CALL-TO-TRUE(0SNCSQL ) RET-A7629568 11:16:33.6310839064 00.0007990156 =007364=
00056 L8004 AP 3180 D2EX1 ENTRY APPLICATION REQUEST - EXEC SQL FETCH RET-80096006 11:16:33.6334723906 00.0023884841 =007364=
00056 L8004 AP 3250 D2D2 ENTRY DB2_API_CALL_CSUB_TOKEN(260A3030) RET-ABCEC542 11:16:33.6334949375 00.0000225468 =007365=
00056 L8004 AP 3251 D2D2 EXIT DB2_API_CALL/OK RET-ABCEC542 11:16:33.6339279375 00.0004330000 =007366=
00056 L8004 AP 3181 D2EX1 EXIT APPLICATION-REQUEST SQLCODE 0 RETURNED ON EXEC SQL FETCH RET-80096006 11:16:33.6339396718 00.00117343 =007367=
00056 L8004 AP 2521 ERM EXIT COBOL-APPLICATION-CALL-TO-TRUE(0SNCSQL ) RET-A7629568 11:16:33.6339789062 00.0007990156 =007368=
```

DB2 CALL ended Time: .0029 sec.

Actual interface to DB2 Time: .0004 secs.

```
00056 QR AP 2520 ERM ENTRY COBOL-APPLICATION-CALL-TO-TRUE(0SNCSQL ) RET-A7629568 11:16:34.6744122031 00.0000560625 =007389=
00056 L8004 AP 3180 D2EX1 ENTRY APPLICATION REQUEST - EXEC SQL FETCH RET-80096006 11:16:34.6758377187 00.0014255156 =007390=
00056 L8004 AP 3250 D2D2 ENTRY DB2_API_CALL_CSUB_TOKEN(260A3030) RET-ABCEC542 11:16:34.6762294375 00.0003917187 =007391=
00056 L8004 AP 3251 D2D2 EXIT DB2_API_CALL/OK RET-ABCEC542 11:16:34.6768985781 00.0006691406 =007392=
00056 L8004 AP 3181 D2EX1 EXIT APPLICATION-REQUEST SQLCODE 0 RETURNED ON EXEC SQL FETCH RET-80096006 11:16:34.6769122812 00.00137031 =007393=
00056 L8004 AP 2521 ERM EXIT COBOL-APPLICATION-CALL-TO-TRUE(0SNCSQL ) RET-A7629568 11:16:34.6769500468 00.0007990156 =007394=
```

DB2 CALL ended Time: .0025 sec.

Actual interface to DB2 Time: .0007 secs.

```
00056 QR AP 2520 ERM ENTRY COBOL-APPLICATION-CALL-TO-TRUE(0SNCSQL ) RET-A7629568 11:16:35.7245992968 00.0001191875 =007411=
00056 L8004 AP 3180 D2EX1 ENTRY APPLICATION REQUEST - EXEC SQL FETCH RET-80096006 11:16:35.7249362500 00.0003369531 =007412=
00056 L8004 AP 3250 D2D2 ENTRY DB2_API_CALL_CSUB_TOKEN(260A3030) RET-ABCEC542 11:16:35.7249644531 00.0000282031 =007413=
00056 L8004 AP 3251 D2D2 EXIT DB2_API_CALL/OK RET-ABCEC542 11:16:35.7267477187 00.0017832656 =007414=
00056 L8004 AP 3181 D2EX1 EXIT APPLICATION-REQUEST SQLCODE 0 RETURNED ON EXEC SQL FETCH RET-80096006 11:16:35.7267789531 00.0012343 =007415=
00056 L8004 AP 2521 ERM EXIT COBOL-APPLICATION-CALL-TO-TRUE(0SNCSQL ) RET-A7629568 11:16:35.7268357812 00.0007990156 =007416=
```

DB2 CALL ended Time: .0022 sec.

Actual interface to DB2 Time: .0018 secs.


Complete your session evaluations online at www.SHARE.org/Seattle-Eval



For the final slide of our THMY transaction we have included three more DB2 calls. This is just to provide you with more examples of the variance in DB2 requests which can occur even when performing the same SQL command, FETCH commands in our example.

- The next few slides represent our transaction THM4, which provides a look at profiling of a normal CICS application program.
- The goal is to not cover all possible situations, but rather to provide you with a method to accomplish this at your sites.

Transaction Profile (THM4) with Trace Entries




Trace Entries for Application profiling:

```

00053 QR AP 00E1 EIP ENTRY ENQ                                REQ(0004) FIELD-A(280160A8 ..y) FIELD-B(09001204 ....)
                                BOUNDARY(0200)                                RET-A7527036 11:46:13.6236505002 00.0000509531 =031515=
00053 QR AP E160 EXEC ENTRY ENQ_RESOURCE('... THEMIS ENQ/DEQ FOR AICA =*' AT X'280182A9') LENGTH(32 AT X'A801A938') NOHANDLE
                                COBOLII STMT_#(00017)                                RET-8008661E 11:46:13.6236679377 00.0000174375 =031516=
00053 QR NQ 0601 NQRN ENTRY INQUIRE_NQRNAME NQRNAME(280182A9 , 00000000 , 00000020) MSG0105(YES)
                                RET-A5CFF6BE 11:46:13.6236834846 00.0000155468 =031517=
00053 QR NQ 0602 NQRN EXIT INQUIRE_NQRNAME/EXCEPTION REASON(NQRNAME_NOT_FOUND) NQRNAME(280182A9 , 00000000 , 00000020) SCOPE()
                                STATE()                                RET-A5CFF6BE 11:46:13.6236987814 00.0000152968 =031518=
00053 QR NQ 0301 NQED ENTRY ENQUEUE_POOL_TOKEN(25A4C500) ENQUEUE_NAME1(280182A9 , 00000020) - ARGUMENT(... THEMIS ENQ/DEQ FOR
                                AICA =*)                                RET-A5CFF7EA 11:46:13.6237066252 00.0000078437 =031519=
00053 QR NQ 0302 NQED EXIT ENQUEUE/OK                                RET-A5CFF7EA 11:46:13.6237164064 00.0000097812 =031520=
00053 QR AP E161 EXEC EXIT ENQ_RESOURCE('... THEMIS ENQ/DEQ FOR AICA =*' AT X'280182A9') LENGTH(32 AT X'A801A938') RESP(0)
                                RESP2(0) NOHANDLE COBOLII STMT_#(00017)    RET-8008889C 11:46:13.6237320314 00.0000156250 =031521=
00053 QR AP 00E1 EIP EXIT ENQ OK                                REQ(00F4) FIELD-A(00000000 ....) FIELD-B(00001204 ....)
                                BOUNDARY(0200)                                RET-A7527036 11:46:13.6237364221 00.0000043906 =031522=
00053 QR AP 00E1 EIP ENTRY READQ_TS                            REQ(0004) FIELD-A(280160A8 ..y) FIELD-B(09000A04 ....)
                                BOUNDARY(0200)                                RET-A7527152 11:46:13.6237461252 00.0000097031 =031523=
00053 QR AP E160 EXEC ENTRY READQ_TS_QUEUE('THEMISTX' AT X'28018240') INTO( AT X'28018248') LENGTH(32 AT X'280182C9') ITEM(1 AT
                                X'A801A938') NOHANDLE COBOLII STMT_#(00017)    RET-8008661E 11:46:13.6274580156 00.0037118903 =031524=
                                RET-8008661A 11:46:13.6274955312 00.0000375156 =031525=
00053 QR AP F704 EITS ENTRY READQ_TS                            RET-8008661A 11:46:13.6275582031 00.0000626718 =031526=
00053 QR AP F705 EITS EXIT READQ_TS RESP=0 RESP2=0
00053 QR AP E161 EXEC EXIT READQ_TS_QUEUE('THEMISTX' AT X'28018240') INTO('>= THEMIS TSQ RECORD FOR AICA =c' AT X'28018248')
                                LENGTH(32 AT X'280182C9') ITEM(1 AT X'A801A938') RESP(0) RESP2(0) NOHANDLE COBOLII STMT_#
                                (00017)                                RET-8008889C 11:46:13.6275857656 00.0000275625 =031527=
00053 QR AP 00E1 EIP EXIT READQ_TS OK                                REQ(00F4) FIELD-A(00000000 ....) FIELD-B(00000A04 ....)
                                BOUNDARY(0200)                                RET-A7527152 11:46:13.6275909062 00.0000051406 =031528=
00053 QR AP 00E1 EIP ENTRY DEQ                                REQ(0004) FIELD-A(280160A8 ..y) FIELD-B(09001206 ....)
                                BOUNDARY(0200)                                RET-A7527246 11:46:13.6276067656 00.0000158593 =031529=
00053 QR AP E160 EXEC ENTRY DEQ_RESOURCE('... THEMIS ENQ/DEQ FOR AICA =*' AT X'280182A9') LENGTH(32 AT X'A801A938') NOHANDLE
                                COBOLII STMT_#(00018)                                RET-8008661E 11:46:13.6276164531 00.0000096875 =031530=

```



Complete your session evaluations online at www.SHARE.org/Seattle-Eval

38

This slide shows another transaction, THM4, which we will use as an example of transaction profiling. We have chosen to begin well into execution of our application program since our goal is to provide a useable technique and not to completely profile our application in this short session.

So the key is to capture key information from either the **EIP ENTRY** or **EIP EXIT** trace entries, not both. In our example we will use the **EIP ENTRY** trace entries to gather the necessary information. For each **EIP ENTRY** we will record the trace sequence number, command being executed, key resource being worked upon, and the **RET-** address back to our program.

For the first command, trace entries **031515** and **031516**, we see an **ENQ** command. The resource is a character string which we obtain the enqueue upon. Plus the return address on trace entry 031515 is **27527036**.

The next command, trace entries **031523** and **031524**, is a **READQ TS** command. The queue name involved is **THEMISTX** and the return address is **27527152** on trace entry 031523.

The last command on this slide, trace entries **031529** and **031530**, is a **DEQ** command. The resource is the same character string from the previous **ENQ** command and the return address is **27527246** on trace entry 031529.

Transaction Profile (THM4) with Trace Entries



Trace Entries for Application profiling (continued):

```

00053 QR NQ 0601 NQRN ENTRY INQUIRE_NQRNAME NQRNAME(2801B2A9 , 00000000 , 00000020) MSG0105(NO)
                                RET-A5CFF9D0 11:46:13.6276307968 00.0000143437 =031531=
00053 QR NQ 0602 NQRN EXIT INQUIRE_NQRNAME/EXCEPTION REASON(NQRNAME_NOT_FOUND) NQRNAME(2801B2A9 , 00000000 , 00000020) SCOPE()
                                STATE()
                                RET-A5CFF9D0 11:46:13.6276459375 00.0000151406 =031532=
00053 QR NQ 0301 NQED ENTRY DEQUEUE POOL_TOKEN(25A4C500) ENQUEUE_NAME1(2801B2A9 , 00000020) - ARGUMENT(... THEMIS ENQ/DEQ FOR
                                AICA =*)
                                RET-A5CFFAE4 11:46:13.6276529531 00.0000070156 =031533=
00053 QR NQ 0302 NQED EXIT DEQUEUE/OK
                                RET-A5CFFAE4 11:46:13.6276614062 00.0000084531 =031534=
00053 QR AP E161 EXEC EXIT DEQ RESOURCE('... THEMIS ENQ/DEQ FOR AICA
                                =*' AT X'2801B2A9') LENGTH(32 AT X'A801A938') RESP()
                                RESP2(0) NOHANDLE COBOLII STMT_#(00018)
                                RET-8008889C 11:46:13.6276760000 00.0000145937 =031535=
00053 QR AP 00E1 EIP EXIT DEQ OK
                                REQ(00F4) FIELD-A(00000000 ...) FIELD-B(00001206 ...)
                                BOUNDARY(0200)
                                RET-A7527246 11:46:13.6276806875 00.0000046875 =031536=
00053 QR AP 00E1 EIP ENTRY ASKTIME-ABSTIME
                                REQ(0004) FIELD-A(280160A8 ...) FIELD-B(09004A02 ...)
                                BOUNDARY(0200)
                                RET-A7526DA0 11:46:13.6276918906 00.0000112031 =031537=
00053 QR AP E160 EXEC ENTRY ASKTIME-ABSTIME( AT X'A801B290') NOHANDLE COBOLII STMT_#(00015)
                                RET-8008661E 11:46:13.6276990781 00.0000071875 =031538=
00053 QR AP E161 EXEC EXIT ASKTIME-ABSTIME( AT X'A801B290') NOHANDLE COBOLII STMT_#(00015)
                                RESP(0) RESP2(0) NOHANDLE COBOLII STMT_#(00015)
                                RET-8008889C 11:46:13.6277737031 00.0000746250 =031539=
00053 QR AP 00E1 EIP EXIT ASKTIME-ABSTIME OK
                                REQ(00F4) FIELD-A(00000000 ...) FIELD-B(00004A02 ...)
                                BOUNDARY(0200)
                                RET-A7526DA0 11:46:13.6277789687 00.0000052656 =031540=
00053 QR AP 00E1 EIP ENTRY FORMATTIME
                                REQ(0004) FIELD-A(280160A8 ...) FIELD-B(09004A04 ...)
                                BOUNDARY(0200)
                                RET-A7526E42 11:46:13.6277880000 00.0000090312 =031541=
00053 QR AP E160 EXEC ENTRY FORMATTIME-ABSTIME(3630138373627 AT X'2801B290') DATE( AT X'2801B298') DATESEP(X'61' AT
                                X'28016260') DAYOFWEEK( AT X'2801B2A8') TIME( AT X'2801B2A0') TIMESEP(X'7A' AT X'A8016288')
                                NOHANDLE COBOLII STMT_#(00016)
                                RET-8008661E 11:46:13.6278048750 00.0000168750 =031542=
00053 QR AP E161 EXEC EXIT FORMATTIME-ABSTIME(3630138373627 AT X'2801B290') DATE('01/13/15' AT X'2801B298') DATESEP(X'61' AT
                                X'28016260') DAYOFWEEK(2 AT X'2801B2A8') TIME('11:46:13' AT X'2801B2A0') TIMESEP(X'7A' AT
                                X'A8016288') RESP(0) RESP2(0) NOHANDLE COBOLII STMT_#(00016)
                                RET-8008889C 11:46:13.6278616562 00.00000567812 =031543=
00053 QR AP 00E1 EIP EXIT FORMATTIME OK
                                REQ(00F4) FIELD-A(00000000 ...) FIELD-B(00004A04 ...)
                                BOUNDARY(0200)
                                RET-A7526F42 11:46:13.6278664687 00.0000048125 =031544=
00053 QR AP 00E1 EIP ENTRY ENQ
                                REQ(0004) FIELD-A(280160A8 ...) FIELD-B(09001204 ...)
                                BOUNDARY(0200)
                                RET-A7527036 11:46:13.6278746093 00.0000081406 =031545=

```

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



39

The first command on this slide, trace entries **031537** and **031538**, is a **ASKTIME-ABSTIME** command. While we cannot see all of the attribute specified, we can see the return address of **27526DA0** on trace entry **031537**.

The next command, trace entries **031541** and **031542**, is a **FORMATTIME** command. This command also does not have any description of its attributes. However, we do have the return address of **27526F42** on trace entry **031541**.

The last command, trace entry **031545**, is a **ENQ** command. The return address on this entry is **27527036**.

Transaction Profile (THM4) with Trace Entries



Trace Entries for Application profiling (continued):

```

00053 QR AP E160 EXEC ENTRY ENQ_RESOURCE('... THEMIS ENQ/DEQ FOR AICA
COBOLII STMT_#(00017) RET-8008661E 11:46:13.6283416875 00.0004670781 =031546=
00053 QR NQ 0601 NQRN ENTRY INQUIRE_NQRNAME NQRNAME(2801B2A9 , 00000000
, 00000020) MSG0105(YES) RET-A5CFF6BE 11:46:13.6283714531 00.0000297656 =031547=
00053 QR NQ 0602 NQRN EXIT INQUIRE_NQRNAME/EXCEPTION REASON(NQRNAME_NOT_FOUND) NQRNAME(2801B2A9 , 00000000 , 00000020) SCOPE()
STATE() RET-A5CFF6BE 11:46:13.6284907812 00.0001193281 =031548=
00053 QR NQ 0301 NQED ENTRY ENQUEUE_POOL_TOKEN(25A4C500) ENQUEUE_NAME1(2801B2A9 , 00000020) - ARGUMENT(... THEMIS ENQ/DEQ FOR
AICA =*) RET-A5CFF7EA 11:46:13.6285032031 00.0000124218 =031549=
00053 QR NQ 0302 NQED EXIT ENQUEUE/OK RET-A5CFF7EA 11:46:13.6285702500 00.0000670468 =031550=
00053 QR AP E161 EXEC EXIT ENQ_RESOURCE('... THEMIS ENQ/DEQ FOR AICA
RESP2(0) NOHANDLE COBOLII STMT_#(00017) RET-8008889C 11:46:13.6285956406 00.0000253906 =031551=
00053 QR AP 00E1 EIP EXIT ENQ OK REQ(00F4) FIELD-A(00000000 ....) FIELD-B(00001204 ....)
BOUNDARY(0200) RET-A7527036 11:46:13.6286008125 00.0000051718 =031552=
00053 QR AP 00E1 EIP ENTRY_READQ_TS REQ(0004) FIELD-A(280160A8 ..y) FIELD-B(09000A04 ....)
BOUNDARY(0200) RET-A7527152 11:46:13.6286153125 00.0000145000 =031553=
00053 QR AP E160 EXEC ENTRY_READQ_TS_QUEUE('THEMISTX' AT X'2801B240') INTO( AT X'2801B248') LENGTH(32 AT X'2801B2C9') ITEM(1 AT
X'A801A938') NOHANDLE COBOLII STMT_#(00017) RET-8008661E 11:46:13.6286287500 00.0000134375 =031554=
00053 QR AP F704 EITS ENTRY READQ_TS RET-80086F1A 11:46:13.6287552187 00.0001264687 =031555=
00053 QR AP F705 EITS EXIT READQ_TS RESP=0 RESP2=0 RET-80086F1A 11:46:13.6288011718 00.0000459531 =031556=
00053 QR AP E161 EXEC EXIT READQ_TS_QUEUE('THEMISTX' AT X'2801B240') INTO('>= THEMIS TSQ RECORD FOR AICA =<' AT X'2801B248')
LENGTH(32 AT X'2801B2C9') ITEM(1 AT X'A801A938') RESP(0) RESP2(0) NOHANDLE COBOLII STMT_#
(00017) RET-8008889C 11:46:13.6288225937 00.0000214218 =031557=
00053 QR AP 00E1 EIP EXIT READQ_TS OK REQ(00F4) FIELD-A(00000000 ....) FIELD-B(00000A04 ....)
BOUNDARY(0200) RET-A7527152 11:46:13.6288273281 00.0000047343 =031558=
00053 QR AP 00E1 EIP ENTRY_DEQ REQ(0004) FIELD-A(280160A8 ..y) FIELD-B(09001206 ....)
BOUNDARY(0200) RET-A7527246 11:46:13.6288384687 00.0000111406 =031559=
00053 QR AP E160 EXEC ENTRY_DEQ_RESOURCE('... THEMIS ENQ/DEQ FOR AICA
COBOLII STMT_#(00018) RET-8008661E 11:46:13.6288470312 00.0000085625 =031560=
00053 QR NQ 0601 NQRN ENTRY INQUIRE_NQRNAME NQRNAME(2801B2A9 , 00000000
, 00000020) MSG0105(NO) RET-A5CFF900 11:46:13.6288583125 00.0000112812 =031561=

```

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



40

The first command, trace entry **031546**, is a **ENQ** command and actually the same one as entry 031545 on the previous slide.

The next command, trace entries **031553** and **031554**, is a **READQ TS** command. The queue name is **THEMISTX** and the return address is **27527152**.

The next command, trace entries **031559** and **031560**, is a **DEQ** command. The resource is the character string found on the previous ENQ command and the return address is **27527246** on trace entry 031559.

Transaction Profile (THM4) with Trace Entries



Trace Entries for Application profiling (continued):

```

00053 QR NQ 0602 NQRN EXIT INQUIRE_NQRNAME/EXCEPTION REASON(NQRNAME_NOT_FOUND) NQRNAME(2801B2A9 , 00000000 , 00000020) SCOPE(
STATE()
RET-A5CFF9D0 11:46:13.6288712187 00.0000129062 =031562=
00053 QR NQ 0301 NQED ENTRY DEQUEUE_POOL_TOKEN(25A4C500) ENQUEUE_NAME1(2801B2A9 , 00000020) - ARGUMENT(... THEMIS ENQ/DEQ FOR
AICA =*)
RET-A5CFFAE4 11:46:13.6288762031 00.0000049843 =031563=
00053 QR NQ 0302 NQED EXIT DEQUEUE/OK
RET-A5CFFAE4 11:46:13.6288835781 00.0000073750 =031564=
00053 QR AP E161 EXEC EXIT DEQ_RESOURCE('... THEMIS ENQ/DEQ FOR AICA =*' AT X'2801B2A9') LENGTH(32 AT X'A801A938') RESP(0)
RET-8008889C 11:46:13.6288976875 00.0000141093 =031565=
00053 QR AP 00E1 EIP EXIT DEQ OK
REQ(00F4) FIELD-A(00000000 ...) FIELD-B(00001206 ...)
RET-A7527246 11:46:13.6289020156 00.0000043281 =031566=
00053 QR AP 00E1 EIP ENTRY ASKTIME-ABSTIME
REQ(0004) FIELD-A(280160A8 ...) FIELD-B(09004A02 ...)
RET-A7526DA0 11:46:13.6289124687 00.0000104531 =031567=
00053 QR AP E160 EXEC ENTRY ASKTIME-ABSTIME( AT X'A801B290') NOHANDLE COBOLII STMT_(00015)
RET-8008661E 11:46:13.6289193593 00.0000068906 =031568=
00053 QR AP E161 EXEC EXIT ASKTIME-ABSTIME(3630138373628 AT X'A801B290') RESP(0) RESP2(0) NOHANDLE COBOLII STMT_(00015)
RET-8008889C 11:46:13.6299431562 00.0010237968 =031569=
00053 QR AP 00E1 EIP EXIT ASKTIME-ABSTIME OK
REQ(00F4) FIELD-A(00000000 ...) FIELD-B(00004A02 ...)
RET-A7526DA0 11:46:13.6299589687 00.0000158125 =031570=
00053 QR AP 00E1 EIP ENTRY FORMATTIME
REQ(0004) FIELD-A(280160A8 ...) FIELD-B(09004A04 ...)
RET-A7526F42 11:46:13.6299734843 00.0000145156 =031571=
00053 QR AP E160 EXEC ENTRY FORMATTIME-ABSTIME(3630138373628 AT X'2801B290') DATE( AT X'2801B298') DATESEP(X'61' AT
X'2801B280') DAYOFWEEK( AT X'2801B2A8') TIME( AT X'2801B2A0') TIMESEP(X'7A' AT X'A8016288')
RET-8008661E 11:46:13.6299951562 00.0000216718 =031572=
00053 QR AP E161 EXEC EXIT FORMATTIME-ABSTIME(3630138373628 AT X'2801B290') DATE('01/13/15' AT X'2801B298') DATESEP(X'61' AT
X'2801B260') DAYOFWEEK(2 AT X'2801B2A8') TIME('11:46:13' AT X'2801B2A0') TIMESEP(X'7A' AT
X'A8016288') RESP(0) RESP2(0) NOHANDLE COBOLII STMT_(00016)
RET-8008889C 11:46:13.6301612812 00.0001661250 =031573=
00053 QR AP 00E1 EIP EXIT FORMATTIME OK
REQ(00F4) FIELD-A(00000000 ...) FIELD-B(00004A04 ...)
RET-A7526F42 11:46:13.6301679062 00.0000066250 =031574=
00053 QR AP 00E1 EIP ENTRY ENQ
REQ(0004) FIELD-A(280160A8 ...) FIELD-B(09001204 ...)
RET-A7527036 11:46:13.6301810156 00.0000131093 =031575=
00053 QR AP E160 EXEC ENTRY ENQ_RESOURCE('... THEMIS ENQ/DEQ FOR AICA =*' AT X'2801B2A9') LENGTH(32 AT X'A801A938') NOHANDLE
COBOLII STMT_(00017)
RET-8008661E 11:46:13.6301911093 00.0000100937 =031576=

```

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



41

The first command, trace entries **031567** and **031568**, is a **ASKTIME-ABSTIME** command. There are no significant attributes shown but the return address is **27526DA0** on trace entry 031567.

The next command, trace entries **031571** and **031572**, is a **FORMATTIME** command. There are no significant attributes on this command either, with a return address of **27526F42** on entry 031571.

The last command, trace entries **031575** and **031576**, is an **ENQ** command. The resource is a character string and the return address is **27527036** on trace entry 031575.

Transaction Profile (THM4) with Trace Entries



Trace Entries for Application profiling (continued):

```
00053 QR NQ 0601 NQRN ENTRY INQUIRE_NQRNAME NQRNAME(2801B2A9 , 00000000 , 00000020) MSG0105(YES) RET-A5CFF6BE 11:46:13.6302048437 00.0000137343 =031577-
00053 QR NQ 0602 NQRN EXIT INQUIRE_NQRNAME/EXCEPTION REASON(NQRNAME_NOT_FOUND) NQRNAME(2801B2A9 , 00000000 , 00000020) SCOPE() RET-A5CFF6BE 11:46:13.6302188281 00.0000139843 =031578=
STATE()
00053 QR NQ 0301 NQED ENTRY ENQUEUE POOL_TOKEN(25A4C500) ENQUEUE_NAME1(2801B2A9 , 00000020) - ARGUMENT(... THEMIS ENQ/DEQ FOR RET-A5CFF7EA 11:46:13.6302269062 00.0000080781 =031579=
AICA =*)
00053 QR NQ 0302 NQED EXIT ENQUEUE/OK RET-A5CFF7EA 11:46:13.6302372968 00.0000103906 =031580=
00053 QR AP E161 EXEC EXIT ENQ RESOURCE('... THEMIS ENQ/DEQ FOR AICA =*' AT X'2801B2A9') LENGTH(32 AT X'A801A938') RESP(0) RET-8008889C 11:46:13.6302529687 00.0000156718 =031581=
RESP2(0) NOHANDLE COBOLII STMT_#(00017)
00053 QR AP 00E1 EIP EXIT ENQ OK REQ(00F4) FIELD-A(00000000 ...) FIELD-B(00001204 ...) RET-A7527036 11:46:13.6302576250 00.0000046562 =031582=
BOUNDARY(0200)
00053 QR AP 00E1 EIP ENTRY_READQ-TS REQ(0004) FIELD-A(280160A8 ...) FIELD-B(09000A04 ...) RET-A7527152 11:46:13.6302663906 00.0000087656 =031583=
BOUNDARY(0200)
00053 QR AP E160 EXEC ENTRY_READQ-TS_QUEUE('THEMISTX' AT X'2801B240') INTO( AT X'2801B248') LENGTH(32 AT X'2801B2C9') ITEM(1 AT X'A801A938') NOHANDLE COBOLII STMT_#(00017) RET-8008661E 11:46:13.6302792968 00.0000129062 =031584=
00053 QR AP 1942 APLI *EXC* RUNAWAY FUNCTION(START_PROGRAM) PROGRAM(THMPGMC) CDFD_STATUS(CDFD) EXECUTION_SET(FULLAPI) RET-8008661E 11:46:13.6302792968 00.0000129062 =031584=
ENVIRONMENT_TYPE(EXEC) SYNCONRETURN(NO) LANGUAGE_BLOCK(276E966C) COMMAREA(00000000 ,
00000000) LINK_LEVEL(2) SYSEIB_REQUEST(NO) RET-A501DE58 11:46:13.6315543750 00.0012750781 =031585=
00053 QR AP 0792 SRP *EXC* RUNAWAY_TASK RET-AS9670C 11:46:13.6316054843 00.0000511093 =031586=
```

We have stopped here in gathering the trace entries since the last one shown indicates that the application has suffered an abend.

It appears that the application has exceeded the amount of allowed processor time without yielding control to CICS.

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



42

The first command, trace entries **031583** and **031584**, is a **READQ TS** command. The queue name is **THEMISTX** and the return address is **27527152** on entry 031583.

Now we see an exception trace entry (***EXC***) at entry number **031585**. This is for a “runaway task” or consumption of too much processor time without yielding control back to CICS. It should be noted at this point that it is possible to suffer an AICA abend even while processing EXEC CICS commands. So long as the commands do not require any form of a wait, the issuing application will gain control immediately and continue with its processing.

The commands **ASKTIME**, **FORMATTIME**, **ENQ** (successful), and **DEQ** do not relinquish control. Only the **ENQ** would cause a wait condition if it fails. The **READQ TS** command does not require a wait if the data items to be processed are already in buffers in memory.

Another command which might not relinquish control is a file control function which the records already reside in buffers in memory. So just because your application is executing commands, does not mean that it will relinquish control so that other applications can run.

Transaction Profile (THM4) with Trace Entries



Trace Entries for Application profiling (continued):

While we were traversing the trace entries we recorded key information about the EXEC CICS commands being issued.

This table reflects our findings, a program loop issuing CICS commands which do not relinquish control to CICS.

	Seq. Number	Command	Resource	RET Address	##
Recorded information AP 00E1 EIP and EXEC Trace points	031515 / 031516	ENQ RESOURCE	Character string	27527036	
	031523 / 031524	READQ-TS	THEMISTX	27527152	
	031529 / 031530	DEQ RESOURCE	Character string	27527246	
	031537 / 031539	ASKTIME-ABSTIME	n/a	27526DA0	
## Remember, The x'80' bit On the address Indicates 31-bit mode	031541 / 031542	FORMATIME	n/a	27526F42	
	031545 / 031546	ENQRESOURCE	Character string	27527036	
	031553 / 031554	READQ-TS	THEMISTX	27527152	
	031559 / 031560	DEQ RESOURCE	Character string	27527246	
	031567 / 031568	ASKTIME-ABSTIME	n/a	27526AD0	
	031571 / 031572	FORMATIME	n/a	27526F42	
	031575 / 031576	ENQ RESOURCE	Character string	27527036	
	031583 / 031584	READQ-TS	THEMISTX	27527152	
	031585	*EXC* Runaway	n/a	n/a	

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



So if you were jotting down the information as we went along, you would end up with a chart like the one on this slide. Stepping back a little to analyze these commands you will notice the loop of ASKTIME, FORMATTIME, ENQ, READQ TS, and DEQ commands. This is evident by the repeat in return addresses involved. Also the same queue name on the READQ TS command and the same character string on the ENQ or DEQ commands. In fact our application programs was in this loop of commands for over 1 1,000 times before failing with the AICA abend.

- The next few slides represent our transaction THM1, which provides a look at the process of task termination.
- While you will see a problem situation as part of this review, we will also expose the sequence of events which occur at time point in the life of a CICS task

Transaction Profile (THM1) with Trace Entries

what happens during normal task termination:

00067 QR	AP 00E1 EIP	ENTRY RETURN	REQ(0004) FIELD-A(27F0A698 .0nq) FIELD-B(09000E08)	
		BOUNDARY(0200)	RET-A751E672 11:16:07.7916224064 00.0000225625	=000699=
00067 QR	AP E160 EXEC	ENTRY RETURN COBOLII STMT_#(00025)	RET-8008661E 11:16:07.7916634221 00.0000410156	=000700=
00067 QR	AP 00E1 EIP	ENTRY FREEMAIN	REQ(0004) FIELD-A(27F01960 .0.-) FIELD-B(09000C04)	
		BOUNDARY(0200)	RET-80544784 11:16:07.7917976408 00.0001342187	=000701=
00067 QR	AP E160 EXEC	ENTRY FREEMAIN DATA(AT X'A7F0F998') SYSEIB NOHANDLE ASM STMT_#(00000404)	RET-8008661E 11:16:07.7918365314 00.0000388906	=000702=
			RET-80086F1A 11:16:07.7918588596 00.0000223281	=000703=
00067 QR	AP E123 EISC	ENTRY FREEMAIN		
00067 QR	SM 0001 SMMF	ENTRY FREEMAIN ADDRESS(27F0F998) CALLER(EXEC) EXEC_KEY(USER)	RET-266C8451 11:16:07.7923888908 00.0005300312	=000704=
			RET-266C8451 11:16:07.7924332346 00.0000443437	=000705=
00067 QR	SM 0002 SMMF	EXIT FREEMAIN/OK USER storage at 27F0F998	RET-80086F1A 11:16:07.7924431564 00.0000099218	=000706=
00067 QR	AP E124 EISC	EXIT FREEMAIN RESP=0 RESP2=0		
00067 QR	AP E161 EXEC	EXIT FREEMAIN DATA(AT X'27F0F998') RESP(0) RESP2(0) SYSEIB NOHANDLE ASM STMT_#(00000404)	RET-8008889C 11:16:07.7926455158 00.0002023593	=000707=
00067 QR	AP 00E1 EIP	EXIT FREEMAIN OK	REQ(00F4) FIELD-A(00000000) FIELD-B(00000C04)	
		BOUNDARY(0200)	RET-80544784 11:16:07.7926576408 00.0000121250	=000708=
00067 QR	AP 00E1 EIP	ENTRY FREEMAIN	REQ(0004) FIELD-A(27F01960 .0.-) FIELD-B(09000C04)	
		BOUNDARY(0200)	RET-80544784 11:16:07.7926964377 00.0000387968	=000709=
00067 QR	AP E160 EXEC	ENTRY FREEMAIN DATA(AT X'A7F0DAD8') SYSEIB NOHANDLE ASM STMT_#(00000404)	RET-8008661E 11:16:07.7927328596 00.0000364218	=000710=
			RET-80086F1A 11:16:07.7927560627 00.0000232031	=000711=
00067 QR	SM 0001 SMMF	ENTRY FREEMAIN ADDRESS(27F0DAD8) CALLER(EXEC) EXEC_KEY(USER)	RET-266C8451 11:16:07.7927676096 00.0000115468	=000712=
			RET-266C8451 11:16:07.7928106877 00.0000430781	=000713=
00067 QR	SM 0002 SMMF	EXIT FREEMAIN/OK USER storage at 27F0DAD8	RET-80086F1A 11:16:07.7928184064 00.0000077187	=000714=
00067 QR	AP E124 EISC	EXIT FREEMAIN RESP=0 RESP2=0		
00067 QR	AP E161 EXEC	EXIT FREEMAIN DATA(AT X'27F0DAD8') RESP(0) RESP2(0) SYSEIB NOHANDLE ASM STMT_#(00000404)	RET-8008889C 11:16:07.7928601564 00.0000417500	=000715=
00067 QR	AP 00E1 EIP	EXIT FREEMAIN OK	REQ(00F4) FIELD-A(00000000) FIELD-B(00000C04)	
		BOUNDARY(0200)	RET-80544784 11:16:07.7930836877 00.0002235312	=000716=
00067 QR	LD 0001 LDLD	ENTRY RELEASE_PROGRAM ENTRY_POINT(A751C000) PROGRAM_TOKEN(00000048_40cc90c0)	RET-A5D366FA 11:16:07.7931512033 00.0000675156	=000717=
			RET-A5D366FA 11:16:07.7931721877 00.0000209843	=000718=
00067 QR	LD 0002 LDLD	EXIT RELEASE_PROGRAM/OK LOAD_POINT(2751C000) PROGRAM_LENGTH(3938) LOCATION(ESDSA)	RET-A5D366FA 11:16:07.7931721877 00.0000209843	=000718=
			RET-A5D3697A 11:16:07.7931938283 00.0000216406	=000719=
00067 QR	PG 1700 PGCH	ENTRY DELETE_OWNED_CHANNELS_SCOPE(TRANSACTION)		

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



45

On this next set of slides we are going to take a look at what happens during task termination. So we begin with trace entry =00699=, which is the **RETURN** command. This is one of the few “ENTRY” trace points for which there is not a corresponding “EXIT” trace point. Again, the **RET-** attribute will reflect the statement in the application program where the command is done.


The only other high-lighted trace entry on this slide is the last line, entry number =000719=, which is a PG domain “ENTRY”. This is the point at which any CHANNEL and CONTAINERS will be deleted on an EXEC CICS RETURN command without the CHANNEL() attribute.

Transaction Profile (THM1) with Trace Entries

What happens during normal task termination (continued):

00067 QR	PG 1800 PGCR	ENTRY DELETE_CONTAINER_POOL	POOL_TOKEN(26F62030)	RET-A508821A 11:16:07.7932057033 00.0000118750	=000720=
00067 QR	PG 1900 PGCR	ENTRY DELETE_CONTAINER_CONTAINER_TOKEN	(26E8F0A0)	CALLER(SYSTEM)	
00067 QR	SM 4201 S2GF	ENTRY FREEMAIN	SUBPOOL_TOKEN(00000048_40B04584 , 00000000_00000071)	ADDRESS(00000048_40D01000)	=000721=
			FREE_LENGTH(1000) REMARK(CSDB) LOCK_POOL(YES) LMLM_ADDRESS(26E5C8C0)		
00067 QR	SM 4202 S2GF	EXIT FREEMAIN/OK		RET-25D40A7F 11:16:07.7932455002 00.0000258437	=000722=
00067 QR	PG 1901 PGCR	EXIT DELETE_CONTAINER/OK		RET-25D40A7F 11:16:07.7932761564 00.0000306562	=000723=
00067 QR	SM 0301 SMGF	ENTRY FREEMAIN	SUBPOOL_TOKEN(35A2A680 , 00000060)	ADDRESS(26F62030) REMARK(CPCR) LOCK_POOL(YES)	=000724=
00067 QR	SM 0302 SMGF	EXIT FREEMAIN/OK		RET-A508A7B0 11:16:07.7932988908 00.0000082812	=000725=
00067 QR	PG 1801 PGCP	EXIT DELETE_CONTAINER_POOL/OK		RET-A508A7B0 11:16:07.7938612502 00.0005623593	=000726=
00067 QR	SM 0301 SMGF	ENTRY FREEMAIN	SUBPOOL_TOKEN(35A2A584 , 0000006C)	ADDRESS(26F61030) FREE_LENGTH(40) REMARK(CHCB)	=000727=
00067 QR	SM 0302 SMGF	EXIT FREEMAIN/OK		RET-A508827E 11:16:07.7940353752 00.0001559687	=000728=
00067 QR	PG 1701 PGCH	EXIT DELETE_OWNED_CHANNELS/OK		RET-A508827E 11:16:07.7940566721 00.0000212968	=000729=
00067 QR	PG 0902 PPGS	EXIT INITIAL_LINK/OK	ABEND_CODE()	RET-A5D3697A 11:16:07.7940636252 00.0000069531	=000730=
00067 QR	XM 131E XMTA	EVENT XM_TXN	state	RET-A5658DAE 11:16:07.7940755002 00.0000118750	=000731=
00067 QR	RM F20B RMUO	ENTRY COMMIT_UOW	CONTINUE(NO)	RET-A562073C 11:16:07.7940833752 00.0000078750	=000732=
00067 QR	RM F20B RMUO	EVENT About_to_call	Client(RMIO) Local_UOW_Id(CE60D16E69E97080)	RET-A5659640 11:16:07.7941001096 00.0000167343	=000733=
00067 QR	AP 2500 ERMSP	ENTRY PERFORM_PREPARE	CONTINUE(NO) WORK_TOKEN(00000000_25A918C8)	RET-A5659640 11:16:07.7941151408 00.0000150312	=000734=
00067 QR	AP 2501 ERMSP	EXIT PERFORM_PREPARE/OK	VOTE(READ_ONLY)	RET-A56FC7EC 11:16:07.7941260939 00.0000109531	=000735=
00067 QR	RM F20B RMUO	EVENT About_to_call	Client(LT) Local_UOW_Id(CE60D16E69E97080)	RET-A56FC7EC 11:16:07.7941319846 00.0000058906	=000736=
00067 QR	RM F20B RMUO	EVENT About_to_call	Client(FC) Local_UOW_Id(CE60D16E69E97080)	RET-A5659640 11:16:07.7941421877 00.0000102031	=000737=
00067 QR	RM F20B RMUO	EVENT About_to_call	Client(APUS) Local_UOW_Id(CE60D16E69E97080)	RET-A5659640 11:16:07.7941569377 00.0000147500	=000738=
00067 QR	RM F20B RMUO	EVENT About_to_call	Client(NQ) Local_UOW_Id(CE60D16E69E97080)	RET-A5659640 11:16:07.7941806252 00.0000236875	=000739=
00067 QR	RM F20B RMUO	EVENT About_to_call	Client(NQ) Local_UOW_Id(CE60D16E69E97080)	RET-A5659640 11:16:07.7941954064 00.0000147812	=000740=


Complete your session evaluations online at www.SHARE.org/Seattle-Eval



This slide begins with the continuation of the CONTAINER cleanup, trace entry numbers =000720=, =000721=, =000724=, and =000727=. We have purposely turned on the PG domain trace entries just to show where the cleanup of the CONTAINERS is performed. Normally we recommend to not include PG domain tracing in the normal scheme of events.

The next high-lighted "ENTRY", number =000733=, represents the start of the process to commit any file or data base updates as part of task termination. Associated with this entry are several "EVENT" trace entries which reflect the coordination with various other CICS components. Here you see the interaction with file control (FC), Enqueue domain (NQ), AP domain (APUS), and we are not sure what (LT) represents.

Transaction Profile (THM1) with Trace Entries



SHARE
Share - Enable - Enhance


What happens during normal task termination (continued):

```

00067 QR  NQ 0201 NQMQ ENTRY PERFORM_PREPARE CONTINUE(NO) WORK_TOKEN(00000000_25A3F680) - ENTRY_WORK_TOKEN(25A3F680) RET-A56FC7EC 11:16:07.7942052346 00.0000098281 =000741=
00067 QR  NQ 0202 NQMQ EXIT PERFORM_PREPARE/OK VOTE(READ_ONLY) - EXIT_WORK_TOKEN(25A3F680) RET-A56FC7EC 11:16:07.7942099064 00.0000046718 -000742=
00067 QR  RM 0351 RMLSP ENTRY Prepare CONTINUE(NO) READONLY(NO) RET-A56F98B8 11:16:07.7942221408 00.0000122343 =000743=
00067 QR  RM 0352 RMLSP EXIT Prepare RESPONSE(OK) CONTINUE(NO) READONLY(NO) VOTE(YES) RET-A56F98B8 11:16:07.7942391408 00.0000170000 -000744=
00067 QR  RM 02E1 RMLSLW ENTRY Write Remark(COMMITTED) RET-A56FA4DC 11:16:07.7942549377 00.0000157968 =000745=
00067 QR  DS 0004 DSSR ENTRY WAIT_MVS_RESOURCE_TYPE(LGWRITE) ECB_ADDRESS(26F1F1A4) PURGEABLE(NO) WLM_WAIT_TYPE(IO)
TEMP_HITGH_PRIORITY(YES) RESOURCE_NAME(DFHLOG)
00067 QR  DS 0005 DSSR EXIT WAIT_MVS/OK RET-A5CACA82 11:16:07.7964143906 00.0021594528 =000746=
00067 QR  SM 0301 SMGF ENTRY GETMAIN SUBPOOL_TOKEN(259B874C , 00000131) SUSPEND(YES) REMARK(L20FL2BL) RET-A5CACA82 11:16:07.7974663437 00.0010519531 =000747=
00067 QR  SM 0302 SMGF EXIT GETMAIN/OK ADDRESS(26F55308) RET-A5CAF290 11:16:07.7974815468 00.0000152031 =000748=
00067 QR  SM 0301 SMGF ENTRY FREEMAIN SUBPOOL_TOKEN(259B874C , 00000131) ADDRESS(26F55098) REMARK(L20FL2BL) RET-A5CAF290 11:16:07.7974890312 00.0000074843 =000749=
00067 QR  SM 0302 SMGF EXIT FREEMAIN/OK RET-A5CAF54E 11:16:07.7975004531 00.0000114218 =000750=
00067 QR  RM 02E2 RMLSLW EXIT Write RESPONSE(OK) RET-A5CAF54E 11:16:07.7975080937 00.0000076406 -000751=
00067 QR  RM 0371 RMLSO ENTRY Commit CONTINUE(NO) VOTE(YES) RESTART(NO) RET-A56FA4DC 11:16:07.7981282968 00.0006202031 =000752=
00067 QR  RM 0378 RMLSO EVENT About_to_call Client(RMI ) Local_UOW_Id(CE60D16E69E97080) RET-A56FASAA 11:16:07.7981792656 00.0000509687 =000753=
00067 QR  AP 2500 ERMSP ENTRY PERFORM_COMMIT RMC_TOKEN(2776347D) CONTINUE(NO) SINGLE_UPDATER(NO) COORDINATOR(NO) INITIATOR(NO)
RESTART(NO) UOW_STATUS(FORWARD) PRESUMPTION(ABORT) RECOVERY_STATUS(UNNECESSARY) TRUE
(EQAQCTRU) RET-A56FASAA 11:16:07.7982629687 00.00000837031 =000754=
00067 QR  AP 2501 ERMSP EXIT PERFORM_COMMIT/OK ACCESSIBLE(YES) FORGET(YES) PASS(YES) ABEND(NO) NEXT_RECOVERY_STATUS(UNNECESSARY)
TRUE(EQAQCTRU) RET-A56D341A 11:16:07.7982715312 00.0000085625 =000755=
00067 QR  RM 0379 RMLSO EVENT Linkset_Is_Empty RET-A56FASAA 11:16:07.7982769062 00.0000053750 =000756=
00067 QR  RM 0372 RMLSO EXIT Commit_RESPONSE(OK) LINKS_PRESENT(NO) RET-A56FASAA 11:16:07.7982866875 00.0000097812 =000757=
00067 QR  RM F20B RMDU EVENT About_to_call Client(RMIO) Local_UOW_Id(CE60D16E69E97080) RET-A56FASAA 11:16:07.7982953125 00.0000086250 =000758=
00067 QR  AP 2500 ERMSP ENTRY PERFORM_COMMIT CONTINUE(NO) UOW_STATUS(FORWARD) WORK_TOKEN(00000000_25A918C8) RET-A5659640 11:16:07.7983017656 00.0000064531 =000759=
00067 QR  AP 2500 ERMSP ENTRY PERFORM_COMMIT CONTINUE(NO) UOW_TOKEN(00000000_25A918C8) RET-A56FC7EC 11:16:07.7983068750 00.0000051093 =000760=

```

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



SHARE
in Seattle 2015

47

The first high-lighted trace entry, number =000746=, represents the writing of the log data as part of the commit process. As you can see, the log records are being written to DFHLOG as indicated by the **RESOURCE_NAME(DFHLOG)** attribute. This is to ensure that these updates are made permanent and will not be rolled back if CICS were to crash and be “emergency restarted”.

The next high-lighted trace entry, number =000754=, indicates the coordination with the Resource Manager Interface (RMI). This would include coordination with **DB2** or **MQ** if necessary.

Finally we see the last high-lighted trace entry, number =000758=, which is the “EXIT” that corresponds to the ENTRY we saw a couple of slides previous. As you can tell, the commit of the protected resources has completed successfully. This is clearly indicated by the **RESPONSE(OK)** attribute.

Transaction Profile (THM1) with Trace Entries

what happens during normal task termination (continued):

00067	QR	AP	2520	ERM	ENTRY	CALL-TRUES-FOR-TASK-END	RET-26798424	11:16:07.7983280625	00.0000211875	=000761=
00067	QR	AP	2521	ERM	EXIT	CALL-TRUES-FOR-TASK-END	RET-26798424	11:16:07.7983446250	00.0000165625	=000762=
00067	QR	AP	2501	ERMSP	EXIT	PERFORM_COMMIT/OK FORGET_RECORD(YES)	RET-A56FC7EC	11:16:07.7983522812	00.0000076562	=000763=
00067	QR	RM	F20B	RMUO	EVENT	About_to_call client(LT) Local_UOW_Id(CE60D16E69E97080)	RET-A5659640	11:16:07.7983594375	00.0000071562	=000764=
00067	QR	XM	1001	XMIQ	ENTRY	SET_TRANSACTION FACILITY_TYPE(NONE) RESTART(NO)	RET-A628D1FA	11:16:07.7983865781	00.0000271406	=000765=
00067	QR	XM	1002	XMIQ	EXIT	SET_TRANSACTION/OK	RET-A628D1FA	11:16:07.7983972343	00.0000106562	=000766=
00067	QR	SM	0E01	SMMC2	ENTRY	FREEMAIN_ALL_TERMINAL TCTTE_ADDRESS(27740570)	RET-A6342FF8	11:16:07.8007805314	00.0023832971	=000767=
00067	QR	SM	0E05	SMMC2	EVENT	Storage_released TERMINAL storage at 26F5D000	RET-A6342FF8	11:16:07.8007895158	00.0000089843	=000768=
00067	QR	SM	0E02	SMMC2	EXIT	FREEMAIN_ALL_TERMINAL/OK	RET-A6342FF8	11:16:07.8008180471	00.0000285312	=000769=
00067	QR	RM	F20B	RMUO	EVENT	About_to_call client(FC) Local_UOW_Id(CE60D16E69E97080)	RET-A5659640	11:16:07.8008500158	00.0000319687	=000770=
00067	QR	RM	F20B	RMUO	EVENT	About_to_call client(APUS) Local_UOW_Id(CE60D16E69E97080)	RET-A5659640	11:16:07.8009003127	00.0000103593	=000772=
00067	QR	RM	F20B	RMUO	EVENT	About_to_call client(NQ) Local_UOW_Id(CE60D16E69E97080)	RET-A5659640	11:16:07.8009003127	00.0000103593	=000772=
00067	QR	NQ	0201	NQMQ	ENTRY	PERFORM_COMMIT CONTINUE(NO) UOW_STATUS(FORWARD) WORK_TOKEN(00000000_25A3F680) - ENTRY_WORK_TOKEN(25A3F680)	RET-A56FC7EC	11:16:07.8009073752	00.0000070625	=000773=
00067	QR	NQ	0202	NQMQ	EXIT	PERFORM_COMMIT/OK FORGET_RECORD(NO) - EXIT_WORK_TOKEN(00000000)	RET-A56FC7EC	11:16:07.8009154533	00.0000080781	=000774=
00067	QR	RM	02D1	RMSLL	ENTRY	Close_Chain	RET-A56FC7EC	11:16:07.8009154533	00.0000080781	=000774=
00067	QR	RM	02D2	RMSLL	EXIT	Close_Chain RESPONSE(OK)	RET-A56F8468	11:16:07.8009276721	00.0000122187	=000775=
00067	QR	RM	F412	RMUO	EXIT	COMMIT_UOW/OK FAILED_LINK(00000000)	RET-A56F8468	11:16:07.8009405314	00.0000128593	=000776=
00067	QR	RM	F412	RMUO	EXIT	COMMIT_UOW/OK FAILED_LINK(00000000)	RET-A5659640	11:16:07.8009474846	00.0000069531	=000777=
XM	QR	PG	0801	PGXM	ENTRY	TERMINATE_TRANSACTION	RET-A565996C	11:16:07.800995471	00.0000520625	=000778=
XM	QR	SM	0301	SMGF	ENTRY	FREEMAIN SUBPOOL_TOKEN(35A2A350 , 00000069) ADDRESS(26F60000) REMARK(HTB) LOCK_POOL(YES)	RET-A5D447D2	11:16:07.8010083596	00.0000088125	=000779=
XM	QR	SM	0302	SMGF	EXIT	FREEMAIN/OK	RET-A5D447D2	11:16:07.8014505002	00.0004481406	=000780=
XM	QR	PG	0802	PGXM	EXIT	TERMINATE_TRANSACTION/OK	RET-A565996C	11:16:07.8014725158	00.0000160156	=000781=
XM	QR	SM	0FD1	SMAR	ENTRY	RELEASE_TRANSACTION_STG	RET-A565996C	11:16:07.8014821721	00.0000096562	=000782=
XM	QR	XM	1001	XMIQ	ENTRY	SET_TRANSACTION STORAGE_VIOLATIONS(INCREMENT)	RET-256454E9	11:16:07.8017422814	00.0002601093	=000783=
XM	QR	XM	1002	XMIQ	EXIT	SET_TRANSACTION/OK	RET-256454E9	11:16:07.8017525939	00.0000103125	=000784=

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



48

Trace entry numbers =000767= and =000769=, represent the freeing of any terminal data areas associated with the task. This would include the last SEND MAP or SEND command output, which should now be displayed on the user's terminal. As we mentioned earlier in this session, it is at this time when CICS checks for storage violations. If the terminal data area were corrupted, we would see the *EXEC* trace entry here.

The next high-lighted trace entry, number =000778=, indicates the point at which the last remnants of the task have been removed from the AP domain. We no longer have an application transaction as far as CICS is concerned.

The last high-lighted trace entry, number =000782=, indicates the start of the process to cleanup all storage acquired by the task during its execution. Again, it is at this point where CICS will check for storage violations. Most applications do not cleanup the storage themselves but rather let CICS perform this function at this point in time.

Transaction Profile (THM1) with Trace Entries

What happens during normal task termination (continued):

```
XM QR SM 0401 SMSR ENTRY INQUIRE_ACCESS ELEMENT_ADDRESS(27F113BF) ELEMENT_LENGTH(1) RET-256457E1 11:16:07.8017816408 00.0000290468 =000785=  
XM QR SM 0402 SMSR EXIT INQUIRE_ACCESS/OK DSA_NAME(EUDSA) ACCESS(USER) RET-256457E1 11:16:07.8017891564 00.0000075156 =000786=  
XM QR SM 0F0C SMAR *EXC* Storage_check_failed_at_address 27F10DB0 FUNCTION(RELEASE_TRANSACTION_STG) RET-A56599AC 11:16:07.8017958908 00.0000067343 =000787=  
XM QR ME 0301 MEME ENTRY SEND_MESSAGE MESSAGE_NUMBER(66) SYSTEM_DUMP_CODE(SM0102) INSERT1(2564625E , 00000002) INSERT2  
(25646240 , 00000008) COMPONENT_ID(SM) RET-2564586F 11:16:07.8023065158 00.0005106250 =000788=
```



During the last several slide we have covered the process of task termination for one of our small test applications.

This is where storage violations are normally detected, after most of the tracks of the task are gone from the CICS region.

This is why trace can be a valuable tool to help you resolve storage violations, especially in larger application environments.

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



49


As you can see by trace entry number **=000787=**, CICS has detected a bas storage area. This is clearly indicated by the ***EXC*** trace entry. Think about what we have just gone through on the past few slides, task termination events as part of the application issues EXEC CICS RETURN command. We have committed file, data base or even MQ messages during this process. Now CICS has encountered a violated storage area and the task is essentially done, no task to abend an roll back at this point.

This is one reason why resolving a storage violation is one of the hardest problems to resolve since the application tracks have already been removed. We are performing this test in a controlled region. Can you imagine this happening in one of your production regions with hundreds if not thousands of task per second going on. We could see the interleaving of a large quantity of trace entries for many different tasks at this point in time.

The last high-lighted trace entry, number **=000788=**, indicates that CICS is about to issue the **"DFHSM0102"** message to indicate the storage violation and take the SVC dump for you to resolve the problem.

- The next few slides represent our transactions THMD and THME, which provides a look potential problems due to concurrent task execution.
- This not meant to be an in depth look at potential problems, but ones which might occur due to operating in a common storage (same address space) environment.

Transaction Profile (THMD and THME) with Trace Entries



SHARE
Share • Connect • Evolve

A look at asynchronous tasks plus a couple of abends:

```

00122 QR  AP 00E1 EIP  ENTRY START                               REQ(0004) FIELD-A(2740A698 , wq) FIELD-B(09001008 , ...)
                                BOUNDARY(0200)                               RET-A9315D2A 09:53:25.4369657971 00.0005023437 =000413=
00122 QR  AP E160 EXEC  ENTRY START TRANSID('THMD' AT X'2740A878') INTERVAL(0 AT X'2740A868') CHANNEL('THEMISCHNL ' AT
                                X'A740FA10') NOHANDLE COBOLII SMT_#(00029) RET-8008661E 09:53:25.4370958908 00.0001300937 =000414=

00122 QR  XM 1101 XMAT  ENTRY ATTACH TRANSACTION_ID(THMD) PRIORITY(0) START_CODE(S) SUSPEND(YES) RETURN_NOT_FOUND(NO)
                                RESTART_COUNT(0) PRIMARY_CLIENT_TYPE(START) PRIMARY_CLIENT_REQ_BLOCK(283812F0 , 00AC6EC4)
                                TRANSACTION_GROUP(SAME) TRANSACTION_GROUP_ID(29312210 , 0000001C) ORIGIN_DATA(29312230 ,
                                000001EC) PASSED_PREV_HOP_COUNT(00000000) ADAPTER_FIELDS(00000000 , 00000000)
                                TRACK_DATA_OVERRIDE(DEFAULT) INITIAL_TS_CURRENT_CTX(NO)
                                RET-A60D28FD 09:53:25.4385227189 00.0001643750 =000440=
                                THMF
                                #122

00122 QR  AP 00E1 EIP  EXIT  START OK                               REQ(00F4) FIELD-A(00000000 , ...) FIELD-B(00001008 , ...)
                                BOUNDARY(0200)                               RET-A9315D2A 09:53:25.4388325627 00.0000562812 =000454=


XM  QR  DS 0012 DSKE  ENTRY TASK_REPLY TASK_TOKEN(26E47800) ATTACH_TOKEN(35A2C080)
                                RET-A5604230 09:53:25.4390069377 00.0000614062 =000458=
XM  QR  XM 1305 XMTA  ENTRY TASK_REPLY USER_TOKEN(25A0A300) TASK_TOKEN(05800020) SUSPEND_TOKEN(05800020)
                                RET-A562073C 09:53:25.4390250939 00.0000181562 =000459=

00123 QR  LD 0001 LDLD  ENTRY ACQUIRE_PROGRAM PROGRAM_TOKEN(00000048_40AC96F0)
                                RET-A5D35A9E 09:53:25.4426565939 00.0000162656 =000509=
00123 QR  LD 0002 LDLD  EXIT  ACQUIRE_PROGRAM/OK ENTRY_POINT(A9313000) LOAD_POINT(29313000) PROGRAM_LENGTH(GEA8)
                                PROGRAM_ATTRIBUTE(REUSABLE) LOCATION(ESDSA) COPY_STATUS(OLD_COPY) FETCH_TIME() LIBRARY
                                (DFHRPL) BUNDLE_INSTALLED_LIB(NO) PRIVATE_LIBRARY(NO)
                                RET-A5D35A9E 09:53:25.4426687814 00.0000121875 =000510=

00123 QR  AP 00E1 EIP  ENTRY ASSIGN                               REQ(0004) FIELD-A(2740A698 , wq) FIELD-B(09000208 , ...)
                                BOUNDARY(0200)                               RET-A9314104 09:53:25.4474197971 00.0000306875 =000588=

00123 QR  AP 00E1 EIP  EXIT  ASSIGN OK                             REQ(00F4) FIELD-A(00000000 , ...) FIELD-B(00000208 , ...)
                                BOUNDARY(0200)                               RET-A9314104 09:53:25.4474871721 00.0000054218 =000591=
                                THMD
                                #123

```



SHARE
in Seattle 2015

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

51

Trace entry numbers =000413= and =000414= show the START command being issued for transaction THMD. Many applications are started asynchronously in CICS, included MQ triggered transactions. Shortly after this event we see trace entry number =000440=, which is CICS actually beginning to execute the transaction. We then see trace entry number =000454=, which is the completion of the START command. All of this being performed by our transaction THMF (task #122).

We now see transaction THMD (task #123) coming to life. The first two trace entries for each CICS application task will be the DSKE and XMTA ENTRY TASK_REPLY, as seen in entry numbers =000458= and =000459= respectively.

Trace entry numbers =000509= and =000510= represent the loading or acquiring of the initial program involved in the transaction. In other trace entries we have omitted you would see the program name, THMPGMA in this case. The importance of these entries is to obtain the ENTRY_POINT() and LOAD_POINT() attributes to allow you to obtain offsets for commands back to the source code.

Finally we see trace entry numbers =000588= and =000591=, which represent the first command our program executed. The command was an ASSIGN TWALENG() SYSID() STARTCODE() in this program.

Transaction Profile (THMD and THME) with Trace Entries

A look at asynchronous tasks plus a couple of abends (continued):

00123 QR	AP 00E1 EIP	ENTRY	ADDRESS BOUNDARY(0200)	REQ(0004) FIELD-A(27A0A698) FIELD-B(09000202) RET-A93141A0 09:53:25.4474955314 00.0000083593 =000592=
00123 QR	AP 00E1 EIP	EXIT	ADDRESS OK BOUNDARY(0200)	REQ(00F4) FIELD-A(00000000) FIELD-B(00000202) RET-A93141A0 09:53:25.4475474221 00.0000052656 =000595=
00123 QR	AP 00E1 EIP	ENTRY	LINK BOUNDARY(0200)	REQ(0004) FIELD-A(27A0A698) FIELD-B(09000E02) RET-A9314BE2 09:53:25.4475589533 00.0000115312 =000596=
00123 QR	AP E160 EXEC	ENTRY	LINK PROGRAM('THMPGMB ' AT X'27A0A818') STMT_#(00019)	CHANNEL('THEMISCHNL ' AT X'A7A0FA10') NOMHANDLE COBOLII RET-8008661E 09:53:25.4483519533 00.0007930000 =000597=
00123 QR	LD 0002 LDLD	EXIT	ACQUIRE_PROGRAM/OK ENTRY_POINT(A931B000) LOAD_POINT(2931B000) PROGRAM_LENGTH(9808) PROGRAM_ATTRIBUTE(REUSABLE) LOCATION(ESDSA) COPY_STATUS(OLD_COPY) FETCH_TIME() LIBRARY (DFHRPL) BUNDLE_INSTALLED_LIB(NO) PRIVATE_LIBRARY(NO) RET-ASD21770 09:53:25.4484224533 00.0000109218 =000600=	
00123 QR	AP 00E1 EIP	ENTRY	GET-CONTAINER BOUNDARY(0200)	REQ(0004) FIELD-A(27A160A8) FIELD-B(09003414) RET-A932073A 09:53:25.4502067500 00.0000960781 =000623=
00123 QR	AP E160 EXEC	ENTRY	GET CONTAINER('THEMISCONT ' AT X'27A1B44B') AT X'27A1BCB4') FLENGTH(32 AT X'27A1B45B')	CHANNEL('THEMISCHNL ' AT X'A7A1B43B') INTO() COBOLII STMT_#(00369) RET-8008661E 09:53:25.4502521718 00.0000454218 =000624=
00123 QR	AP 00E1 EIP	EXIT	GET-CONTAINER OK BOUNDARY(0200)	REQ(00F4) FIELD-A(00000000) FIELD-B(00003414) RET-A932073A 09:53:25.4503817031 00.0000063281 =000632=
00123 QR	AP 00E1 EIP	ENTRY	XCTL BOUNDARY(0200)	REQ(0004) FIELD-A(27A160A8) FIELD-B(09000E04) RET-A9320C3A 09:53:25.4503919062 00.0000102031 =000633=
00123 QR	AP E160 EXEC	ENTRY	XCTL PROGRAM('THMPGMC ' AT X'27A16250') STMT_#(00372)	CHANNEL('THEMISCHNL ' AT X'A7A1B43B') NOMHANDLE COBOLII RET-8008661E 09:53:25.4504241875 00.0000322812 =000634=
00123 QR	LD 0002 LDLD	EXIT	ACQUIRE_PROGRAM/OK ENTRY_POINT(A9324810) LOAD_POINT(29324810) PROGRAM_LENGTH(3948) PROGRAM_ATTRIBUTE(REUSABLE) LOCATION(ESDSA) COPY_STATUS(OLD_COPY) FETCH_TIME() LIBRARY (DFHRPL) BUNDLE_INSTALLED_LIB(NO) PRIVATE_LIBRARY(NO) RET-ASD81660 09:53:25.4505853125 00.0000127343 =000637=	

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



52

Transaction THMD continues execution with the command ADDRESS TWA() as seen in trace entry numbers =000592= and =000595=.

The next command is LINK PROGRAM() CHANNEL() as seen in entry numbers =000596= and =000597=. The program being invoked is THMPGMB, with channel THEMISCHNL.


As with any program invocation, we will be able to obtain the ENTRY_POINT() and LOAD_POINT() in an LD domain trace entry as seen in number =000600=.

We have omitted many trace entries for program THMPGMB initializing to show the first command it issued. The GET CONTAINER command seen in entry numbers =000623=, =000624=, and =000632=.

Next the program issues an XCTL PROGRAM() CHANNEL() command to pass control to THMPGMC as seen in trace entry numbers =000633= and =000634=.

Finally on this slide is the LD domain trace entry number =000637= where we can obtain the ENTRY_POINT() and LOAD_POINT() for THMPGMC.


Transaction Profile (THMD and THME) with Trace Entries


SHARE
Share • Support • Success

A look at asynchronous tasks plus a couple of abends (continued):

00123 QR	AP 00E1 EIP	ENTRY ASSIGN	BOUNDARY(0200)	REQ(0004) FIELD-A(27A160A8 ..-y) FIELD-B(09000208 ..-y) RET-A93250F0 09:53:25.4539236718 00.0000289218 =000685=
00123 QR	AP 00E1 EIP	EXIT	ASSIGN OK BOUNDARY(0200)	REQ(00F4) FIELD-A(00000000 ..-y) FIELD-B(00000208 ..-y) RET-A93250F0 09:53:25.4539797656 00.0000060312 =000688=
00123 QR	AP 00E1 EIP	ENTRY	ADDRESS BOUNDARY(0200)	REQ(0004) FIELD-A(27A160A8 ..-y) FIELD-B(09000202 ..-y) RET-A93251EE 09:53:25.4539883125 00.0000085468 =000689=
00123 QR	AP 00E1 EIP	EXIT	ADDRESS OK BOUNDARY(0200)	REQ(00F4) FIELD-A(00000000 ..-y) FIELD-B(00000202 ..-y) RET-A93251FE 09:53:25.4550637031 00.0000071718 =000692=
00123 QR	AP 00E1 EIP	ENTRY	GET-CONTAINER BOUNDARY(0200)	REQ(0004) FIELD-A(27A160A8 ..-y) FIELD-B(090003414 ..-y) RET-A9325474 09:53:25.4550771093 00.0000134062 =000693=
00123 QR	AP E160 EXEC	ENTRY	GET CONTAINER('THEMISCONT ' AT X'27A1B345') CHANNEL('THEMISCHNL ' AT X'A7A1B335') INTO(AT X'27A1B360') FLENGTH(32 AT X'27A1B355') COBOLII STMT_#(00009)	RET-8008661E 09:53:25.4551228281 00.0000457187 =000694=
00123 QR	AP 00E1 EIP	EXIT	GET-CONTAINER OK BOUNDARY(0200)	REQ(00F4) FIELD-A(00000000 ..-y) FIELD-B(00003414 ..-y) RET-A9325474 09:53:25.4552430312 00.0000061875 =000702=
00123 QR	AP 00E1 EIP	ENTRY	GETMAIN BOUNDARY(0200)	REQ(0004) FIELD-A(27A160A8 ..-y) FIELD-B(09000C02 ..-y) RET-A9326490 09:53:25.4552536875 00.0000106562 =000703=
00123 QR	AP E160 EXEC	ENTRY	GETMAIN SET(AT X'27A173EC') FLENGTH(512 AT X'A7A16210') COBOLII STMT_#(00019)	RET-8008661E 09:53:25.4552792968 00.0000256093 =000704=
00123 QR	SM 0C02 SMMG	EXIT	GETMAIN/OK ADDRESS(27A1C3A8)	RET-266C6EA3 09:53:25.4553266093 00.0000137031 =000707=
00123 QR	AP E161 EXEC	EXIT	GETMAIN SET(X'27A1C3A8' AT X'27A173EC') FLENGTH(512 AT X'A7A16210') RESP(0) RESP2(0) COBOLII STMT_#(00019)	RET-8008889C 09:53:25.4553636875 00.0000316718 =000709=
00123 QR	AP 00E1 EIP	EXIT	GETMAIN OK BOUNDARY(0200)	REQ(00F4) FIELD-A(00000000 ..-y) FIELD-B(00000C02 ..-y) RET-A9326490 09:53:25.4553697343 00.0000060468 =000710=
00123 QR	AP 00E1 EIP	ENTRY	GETMAIN BOUNDARY(0200)	REQ(0004) FIELD-A(27A160A8 ..-y) FIELD-B(09000C02 ..-y) RET-A9326564 09:53:25.4564361718 00.0010664375 =000711=
00123 QR	AP E160 EXEC	ENTRY	GETMAIN SET(AT X'27A173F0') FLENGTH(512 AT X'A7A16210') CICSDATAKEY COBOLII STMT_#(00020)	RET-8008661E 09:53:25.4564743125 00.0000381406 =000712=

Complete your session evaluations online at www.SHARE.org/Seattle-Eval


SHARE
in Seattle 2015

53


The first command executed by THMPGMC is an ASSIGN CWALENG() as seen in trace entry numbers =000685= and =000688=. This is followed by the ADDRESS CWA() command as seen in entry numbers =000698= and =000692=.

The next command performed is a GET CONTAINER for container **THEMISCONT** as seen in entry numbers =000693=, =000694=, and =000702=.

The execution of a GETMAIN command for **512 bytes** of storage is next, entry numbers =000703=, =000704=, =000707=, =000709=, and =000710=. The address of the acquired storage (**27A1C3A8**) can be found in both entry numbers =000707= and =000709=. At times it may be necessary to determine the actual owner of storage when working with a difficult storage violation problem.

A second GETMAIN command for **512 bytes** of **CICSDATAKEY** storage is performed as seen in entry numbers =000711= and =000712=.

Transaction Profile (THMD and THME) with Trace Entries



A look at asynchronous tasks plus a couple of abends (continued):

```


00123 QR SM 0C02 SMMG EXIT GETMAIN/OK ADDRESS(2943A008) RET-266C6EA3 09:53:25.4565470156 00.0000322031 =000715=
00123 QR AP E161 EXEC EXIT GETMAIN SET(X'2943A008' AT X'27A173F0') FLENGTH(S12 AT X'A7A16210') CICSDATAKEY RESP(0) RESP2(0)
COBOLII STMT_#(00020) RET-8008889C 09:53:25.4565877968 00.0000346250 =000717=
00123 QR AP 00E1 EIP EXIT GETMAIN OK REQ(00F4) FIELD-A(00000000 ...) FIELD-B(00000C02 ...)
BOUNDARY(0200) RET-A9326564 09:53:25.4565944375 00.0000066406 =000718=
00122 QR AP 00E1 EIP ENTRY START REQ(0004) FIELD-A(2740A698 . WQ) FIELD-B(09001008 ...)
BOUNDARY(0200) RET-A931603A 09:53:26.4839946408 00.0000293125 =000735=
00122 QR AP E160 EXEC ENTRY START TRANSID('THME' AT X'2740A878') INTERVAL(0 AT X'2740A868') CHANNEL('THEMISCHNL ' AT
X'A740FA10') NOWHANDLE COBOLII STMT_#(00030) RET-8008661E 09:53:26.4848156721 00.0008210312 =000736=
00122 QR XM 1101 XMAT ENTRY ATTACH TRANSACTION_ID(THME) PRIORITY(0) START_CODE(S) SUSPEND(YES) RETURN_NOT_FOUND(NO)
RESTART_COUNT(0) PRIMARY_CLIENT_TYPE(START) PRIMARY_CLIENT_REQ_BLOCK(28381660 , 00AC6EC4)
TRANSACTION_GROUP(SAME) TRANSACTION_GROUP_ID(29312710 , 0000001C) ORIGIN_DATA(29312230 ,
000001EC) PASSED_PREV_HOP_COUNT(00000000) ADAPTER_FIELDS(00000000 , 00000000)
TRACK_DATA_OVERRIDE(DEFAULT) INITIAL_IS_CURRENT_CTX(NO)
RET-A60D28F0 09:53:26.4870175783 00.0003847500 =000762=
00122 QR AP 00E1 EIP EXIT START OK REQ(00F4) FIELD-A(00000000 ...) FIELD-B(00001008 ...)
BOUNDARY(0200) RET-A931603A 09:53:26.4874997502 00.0000102812 =000776=
XM QR DS 0012 DSKE ENTRY TASK_REPLY TASK_TOKEN(26E59800) ATTACH_TOKEN(35A2C680)
RET-A560423D 09:53:26.4877482346 00.0000842968 =000780=
XM QR XM 1305 XMTA ENTRY TASK_REPLY USER_TOKEN(28379B00) TASK_TOKEN(05880009) SUSPEND_TOKEN(05880009)
RET-A562073C 09:53:26.4877684533 00.0000202187 =000781=
00124 QR LD 0001 LDLD ENTRY ACQUIRE_PROGRAM PROGRAM_TOKEN(00000048_40AC96F0)
RET-ASD35A9E 09:53:26.4964467968 00.0001416875 =000829=
00124 QR LD 0002 LDLD EXIT ACQUIRE_PROGRAM/OK ENTRY_POINT(A9313000) LOAD_POINT(29313000) PROGRAM_LENGTH(6EAB)
PROGRAM_ATTRIBUTE(REUSABLE) LOCATION(ESDSA) COPY_STATUS(OLD_COPY) FETCH_TIME() LIBRARY
(OFHRLPL) BUNDLE_INSTALLED_LIB(NO) PRIVATE_LIBRARY(NO)
RET-ASD35A9E 09:53:26.4964706875 00.0000238906 =000830=

```

THMF
#122

THME
#124

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



54

On the top of this slide we see the end of the GETMAIN command from the previous slide in entry numbers =000715=, =000717=, and =000718=. Again the acquired storage address (2943A008) can be seen in entry numbers =000715= and =000717=. After this, THMD relinquished control due to a command which caused a wait condition.

We then see THMF (task #122) gain control once again and execute another START command for transaction THME. All of the trace entries in this small section are from that START command. The task also performs a function which causes THMF to relinquish control back to CICS.

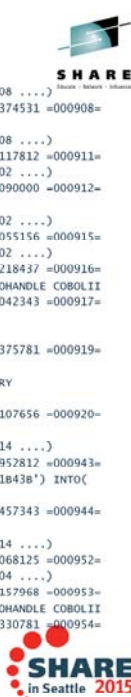
Next we see the beginning of task #124, transaction THME as indicated by entry numbers =000780= and =000781=. We have omitted many trace entries from the initiation of the task to show the LD domain entries, numbers =000829= and =000830=, for program THMPGMA which is the initial program for the transaction..

Transaction Profile (THMD and THME) with Trace Entries

A look at asynchronous tasks plus a couple of abends (continued):

00124 QR	AP 00E1 EIP	ENTRY ASSIGN BOUNDARY(0200)	REQ(0004) FIELD-A(27E0A698 \wq) FIELD-B(09000208) RET-A9314104 09:53:26.5011330781 00.0000374531 =000908=
00124 QR	AP 00E1 EIP	EXIT ASSIGN OK BOUNDARY(0200)	REQ(00F4) FIELD-A(00000000) FIELD-B(00000208) RET-A9314104 09:53:26.5012089531 00.0000117812 =000911=
00124 QR	AP 00E1 EIP	ENTRY ADDRESS BOUNDARY(0200)	REQ(0004) FIELD-A(27E0A698 \wq) FIELD-B(09000202) RET-A93141A0 09:53:26.5012179531 00.0000090000 =000912=
00124 QR	AP 00E1 EIP	EXIT ADDRESS OK BOUNDARY(0200)	REQ(00F4) FIELD-A(00000000) FIELD-B(00000202) RET-A93141A0 09:53:26.5012734687 00.0000055156 =000915=
00124 QR	AP 00E1 EIP	ENTRY LINK BOUNDARY(0200)	REQ(0004) FIELD-A(27E0A698 \wq) FIELD-B(09000E02) RET-A9314D0A 09:53:26.5012953125 00.0000218437 =000916=
00124 QR	AP E160 EXEC	ENTRY LINK PROGRAM('THMPGMB ' AT X'27E0A818') CHANNEL('THEMISCHNL ' AT X'A7E0FA10') NOHANDLE COBOLII STMT_#(00020)	RET-8008661E 09:53:26.5018995468 00.0006042343 =000917=
00124 QR	LD 0001 LDLD	ENTRY ACQUIRE_PROGRAM PROGRAM_TOKEN(00000048_40AC97F8)	RET-A5D21770 09:53:26.5021227187 00.0000375781 =000919=
00124 QR	LD 0002 LDLD	EXIT ACQUIRE_PROGRAM/OK ENTRY_POINT(A931B000) LOAD_POINT(2931B000) PROGRAM_LENGTH(9808) PROGRAM_ATTRIBUTE(REUSABLE) LOCATION(ESDSA) COPY_STATUS(OLD_COPY) FETCH_TIME() LIBRARY (DFHRPL) BUNDLE_INSTALLED_LIB(NO) PRIVATE_LIBRARY(NO)	RET-A5D21770 09:53:26.5021334843 00.0000107656 =000920=
00124 QR	AP 00E1 EIP	ENTRY GET-CONTAINER BOUNDARY(0200)	REQ(0004) FIELD-A(27E160A8 ..-y) FIELD-B(09003414) RET-A932073A 09:53:26.5100314533 00.0000952812 =000943=
00124 QR	AP E160 EXEC	ENTRY GET CONTAINER('THEMISCONT ' AT X'27E1B448') CHANNEL('THEMISCHNL ' AT X'A7E1B43B') INTO(AT X'27E1B448') FLENGTH(32 AT X'27E1B45B') COBOLII STMT_#(00369)	RET-8008661E 09:53:26.5100771877 00.0000457343 =000944=
00124 QR	AP 00E1 EIP	EXIT GET-CONTAINER OK BOUNDARY(0200)	REQ(00F4) FIELD-A(00000000) FIELD-B(00003414) RET-A932073A 09:53:26.5102956721 00.0000068125 =000952=
00124 QR	AP 00E1 EIP	ENTRY XCTL BOUNDARY(0200)	REQ(0004) FIELD-A(27E160A8 ..-y) FIELD-B(09000E04) RET-A9320D1A 09:53:26.5103114689 00.0000157968 =000953=
00124 QR	AP E160 EXEC	ENTRY XCTL PROGRAM('THMPGMC ' AT X'27E16250') CHANNEL('THEMISCHNL ' AT X'A7E1B43B') NOHANDLE COBOLII STMT_#(00373)	RET-8008661E 09:53:26.5103445471 00.0000330781 =000954=

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



On this slide we see transaction THME executing its first command, an ASSIGN TWALENG() SYSID() STARTCODE(). This can be seen in entry numbers =000908= and =000911=.

Transaction THME continues execution with the command ADDRESS TWA() as seen in trace entry numbers =000912= and =000915=.

The next command is LINK PROGRAM() CHANNEL() as seen in entry numbers =000916= and =000917=. The program being invoked is THMPGMB, with channel THEMISCHNL.

As part of the LINK command we see the LD domain entry numbers =000919= and =000920=, which will provide us with the ENTRY_POINT() and LOAD_POINT().

We have omitted many trace entries for program THMPGMB initializing to show the first command it issued. The GET CONTAINER command seen in entry numbers =000943=, =000944=, and =000952=.


Next the program issues an XCTL PROGRAM() CHANNEL() command to pass control to THMPGMC as seen in trace entry numbers =000953= and =000954=.

Transaction Profile (THMD and THME) with Trace Entries

A look at asynchronous tasks plus a couple of abends (continued):

00124	QR	LD 0001	LDLD	ENTRY	ACQUIRE_PROGRAM PROGRAM_TOKEN(00000048_40AC9900)	RET-ASD81660 09:53:26.5107559377 00.0000302812 =000956=
00124	QR	LD 0002	LDLD	EXIT	ACQUIRE_PROGRAM/OK ENTRY_POINT(A9324810) LOAD_POINT(29324810) PROGRAM_LENGTH(3948) PROGRAM_ATTRIBUTE(REUSABLE) LOCATION(ESDSA) COPY_STATUS(OLD_COPY) FETCH_TIME() LIBRARY (DFHRPL) BUNDLE_INSTALLED_LIB(NO) PRIVATE_LIBRARY(NO)	RET-ASD81660 09:53:26.5107700783 00.0000141406 =000957=
00124	QR	AP 00E1	EIP	ENTRY	ASSIGN BOUNDARY(0200)	REQ(0004) FIELD-A(27E160A8 ..-y) FIELD-B(09000208) RET-A93250F0 09:53:26.5139431875 00.0000295312 =001005=
00124	QR	AP 00E1	EIP	EXIT	ASSIGN OK BOUNDARY(0200)	REQ(00F4) FIELD-A(00000000) FIELD-B(00000208) RET-A93250F0 09:53:26.5140145625 00.0000160156 =001008=
00124	QR	AP 00E1	EIP	ENTRY	ADDRESS BOUNDARY(0200)	REQ(0004) FIELD-A(27E160A8 ..-y) FIELD-B(09000202) RET-A93251EE 09:53:26.5140254531 00.0000108906 =001009=
00124	QR	AP 00E1	EIP	EXIT	ADDRESS OK BOUNDARY(0200)	REQ(00F4) FIELD-A(00000000) FIELD-B(00000202) RET-A93251EE 09:53:26.5150350781 00.0000076406 =001012=
00124	QR	AP 00E1	EIP	ENTRY	GET-CONTAINER BOUNDARY(0200)	REQ(0004) FIELD-A(27E160A8 ..-y) FIELD-B(09003414) RET-A9325474 09:53:26.5150629531 00.0000278750 =001013=
00124	QR	AP E160	EXEC	ENTRY	GET CONTAINER(*THEMISCONT ' AT X'27E18345') CHANNEL(*THEMISCHNL ' AT X'7E18335') INTO(AT X'27E18360') FLENGTH(32 AT X'27E18355') COBOLII STMT_#(00009)	RET-8008661E 09:53:26.5151130156 00.0000500625 =001014=
00124	QR	AP 00E1	EIP	EXIT	GET-CONTAINER OK BOUNDARY(0200)	REQ(00F4) FIELD-A(00000000) FIELD-B(00003414) RET-A9325474 09:53:26.5152607656 00.0000067968 =001022=
00124	QR	AP 1942	APLI	*EXC*	Program-Check FUNCTION(START_PROGRAM) PROGRAM(THMPGMC) CDFP_STATUS(CDFP) EXECUTION_SET(FULLAPI) ENVIRONMENT_TYPE(EXEC) SYNCONRETURN(NO) LANGUAGE_BLOCK(283FE8CC) COMMAREA(00000000 , 00000000) LINK_LEVEL(2) SYSEIB_REQUEST(NO)	RET-A5D1DE58 09:53:26.6081184064 00.0928576408* =001023=
00124	QR	ME 0501	MEIN	ENTRY	INQUIRE_MESSAGE_DATA MODULE_PTR(00000048_40200000) MODULE_NAME(DFHMETIE) MESSAGE_NUMBER(26E) ROUTE_CODES_BUFFER(299823BD , 00000000 , 0000001C) QUEUES_BUFFER(29982357 , 00000000 , 00000019) MESSAGE_PREFIX(DFHSR0622)	RET-25680867 09:53:26.6613009064 00.0000179062 =001029=
00124	QR	AP 0781	SRP	*EXC*	ABEND_ASRA PROGRAM(THMPGMC) OFFSET(00001F3C) EXEC_KEY(USER) OC4_ON(EUDSA)	RET-AS48870C 09:53:34.3216402033 00.0000078437 =001040=

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



As part of the XCTL command we see the LD domain entry numbers =000956= and =000957=, which will provide us with the **ENTRY_POINT()** and **LOAD_POINT()**.

We have omitted many trace entries for program THMPGMC initializing to show the first command it issued, an ASSIGN CWALENG command. This is indicated by entry numbers =001005= and =001008=. This is followed by an ADDRESS CWA() command as seen in entry numbers =001009= and =001012=.


We then see a GET CONTAINER command being issued as seen in entry numbers =001013=, =001014=, and =001022=. This is for container **THEMISCONT**, which has a length of **32 bytes**.

Immediately after the GET CONTAINER command we see that transaction THME (task #124) causes a **"Program-check"** as indicated by entry numbers =001023= and =001024=. Since this example is showing short format trace entries we do not see a lot of information about the abend. We will next have a look at their full trace counterparts where a bit more information is provided.

The next trace entry on this slide, number =001029=, provides us with the message number (**DFHSR0622**) and dump code (**SR0622**) being issued by CICS. This is followed by another ***EXC*** exception trace entry, number =001040=. We will also look at its full trace entry counterpart to see what more information is provided.

Transaction Profile (THMD and THME) with Trace Entries

A look at asynchronous tasks plus a couple of abends (continued):




```

AP 1942 APLI *EXC* - Program-Check FUNCTION(START_PROGRAM) PROGRAM(THMPGMC) CEDF_STATUS(CEDF) EXECUTION_SET(FULLAPI)
ENVIRONMENT_TYPE(EXEC) SYNCONRETURN(NO) LANGUAGE_BLOCK(283FE8CC) COMMAREA(00000000 , 00000000) LINK_LEVEL(2)
SYSEIB_REQUEST(NO)
TASK-00124 KE_NUM-006D TC8-QR /008E6968 RET-ASD1DE58 TIME-09:53:26.6081184064 INTERVAL-00.0928576408* =001023=
1-0000 01880000 000000DA 00000000 00000000 B81B4750 00000000 02000100 E3C8D4D7 *.....&.....THMP*
.....
01A0 00000000 00000000 84000000 00000000 00000000 256B215E .....:.....*
2-0000 F0C3F461 C1D2C5C1 018600C4 00000000 C4C6C8C1 D7D3C9F1 00000000 26204500 *OC4/AKEA.F.D...DFHAPL11.....*
.....
02C0 00000000 27A1C400 00000000 00000000 .....D.....*

AP 0781 SRP *EXC* - ABEND_ASRA PROGRAM(THMPGMC ) OFFSET(00001F3C) EXEC_KEY(USER) OC4_ON(EUDSA)
TASK-00124 KE_NUM-006D TC8-QR /008E6968 RET-AS48870C TIME-09:53:34.3216402033 INTERVAL-00.0000078437 =001040=
1-0000 E3C8D4D7 C7D4C340 F0C3F461 C1D2C5C1 00001F3C 00020781 *THMPGMC OC4/AKEA.....a *
2-0000 05 *.....*
3-0000 F0C3F461 C1D2C5C1 018600C4 00000000 C4C6C8C1 D7D3C9F1 00000000 26204500 *OC4/AKEA.F.D...DFHAPL11.....*
0020 00000000 25A51000 00000000 26E59800 35A2C680 0000001B 00000010 FFFFFFFF *.....Vq...5F.....*
0040 07952000 80000000 00000000 2932674C 00020010 27A1C400 00000000 00000000 *.....<.....D.....*
0060 90800000 00000000 00000000 27E173F8 00000000 27E161D0 00000000 27A1C3A8 *.....8...../.....Cy*
0080 00000000 00000200 00000000 27A1C3A8 00000000 26E67798 00000000 26E67750 *.....Cy...W.q...W.&*
00A0 00000000 5C000000 00000000 27E1A5A0 00000000 27E17298 00000000 29324950 *.....V.....*
00C0 00000000 29325E74 00000000 29324900 00000000 27E160A8 00000000 A932673C *.....y...z...*
00E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....<.....D.....*
0100 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0120 00000000 01000002 07952000 80000000 00000000 2932674C 00020010 27A1C400 *.....<.....D.....*
0140 00000000 00000000 90800000 00000000 00000000 27E173F8 00000000 27E161D0 *.....8...../.....}.....*
0160 00000000 27A1C3A8 00000000 00000200 00000000 27A1C3A8 00000000 26E67798 *.....Cy...W.q...*
0180 00000000 26E67750 00000000 5C000000 00000000 27E1A5A0 00000000 27E17298 *.....W.&...V.....v.....q*
01A0 00000000 29324950 00000000 29325E74 00000000 29324900 00000000 27E160A8 *.....&.....v.....y*
01C0 00000000 A932673C 00000000 00000000 00000000 00000000 00000000 00000000 *.....z.....*
01E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0200 00000000 00000000 00000000 01000002 CE6F00B1 5CD5A881 00000000 00000000 *.....7.3*Nya.....*
0220 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0240 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0260 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0280 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
02A0 +5830E18 00000001 00010005 C0000000 00000000 27E1748E 25843200 0000124C *.....C.....{.....D.....*
02C0 00000000 27A1C400 00000000 00000000 .....D.....*

```

Complete your session evaluations online at www.SHARE.org/Seattle-Eval


57

The first entry, number =001023=, is the full trace entry corresponding to the short trace entry of the same number on the previous slide. Here we see a couple of data areas with data area 1- being the parameter list, which is not fully documented. Data area 2- is the Kernel Error Data table entry for the task, which is document in the CICS Data Areas manual. We will describe it a little bit with the next full trace entry.

The second full trace entry, number =001040=, corresponds to the short entry of the same number on the previous slide. This entry is documented in the CICS Trace Entries manual. First have a quick look at the attributes provided, **PROGRAM(THMPGMC)**, **OFFSET(00001F3C)**, and **OC4_ON(EUDSA)**. From these we gain knowledge of the offending program, plus the offset to the instruction within the program. We can also see that the problem is in accessing a storage area in the EUDSA which we do not have addressability to, as indicated by the OC4_ON(EUDSA) attribute.


Data area 1- is labeled as **TCASRDAT** for trace point AP 0781 in the CICS Trace Entries manual. This provides us with the program name, KE error code, program offset, and some flags. These are what we see as the interpreted attributes mentioned above. Data area 2- is labeled as TCAPCSTG or the storage hit indicator. Data area 3- is the Kernel Error Data table entry. This is fully documented in the CICS Data Areas manual. However a few key offsets are:

- +0 – which is the System and User abend codes
- +40 – which is the PSW at the time of the abend
- +68 – which is the saved registers (8 bytes each) at the time of the abend

While not clearly indicated, with a little research you would be able to determine that THME (task #124) attempted to modify storage of another task. Both transactions are defined with ISOLATE(YES) and TRANISO(YES) has been specified.

Transaction Profile (THMD and THME) with Trace Entries

A look at asynchronous tasks plus a couple of abends (continued):



```

00123 QR  AP 1942 APLI *EXC* Program-Check  FUNCTION(START_PROGRAM) PROGRAM(THMPGMC) CEDF_STATUS(CEDF) EXECUTION_SET(FULLAPI)
ENVIRONMENT_TYPE(EXEC) SYNCONRETURN(NO) LANGUAGE_BLOCK(283FE8CC) COMMAREA(00000000 ,
00000000) LINK_LEVEL(2) SYSEIB_REQUEST(NO) RET-ASD1DE58 09:53:34.4935717033 00.0601490158* =001111=


00123 QR  ME 0501 MEIN ENTRY  INQUIRE_MESSAGE_DATA MODULE_PTR(00000048_40200000) MODULE_NAME(DFHMET1E) MESSAGE_NUMBER(26E)
ROUTE_CODES_BUFFER(2942D3BD , 00000000 , 0000001C) QUEUES_BUFFER(2942D357 , 00000000 ,
00000019) MESSAGE_PREFIX(DFHSR0622) RET-25680B67 09:53:34.4936957346 00.0000174531 =001116=


00123 QR  AP 0781 SRP *EXC* ABEND_ASRA PROGRAM(THMPGMC ) OFFSET(00001E7C) EXEC_KEY(USER) 0C4_ON(ECD5A)
RET-ASAB870C 09:53:45.9127418593 00.0000077187 =001127=

AP 1942 APLI *EXC* - Program-Check  FUNCTION(START_PROGRAM) PROGRAM(THMPGMC) CEDF_STATUS(CEDF) EXECUTION_SET(FULLAPI)
ENVIRONMENT_TYPE(EXEC) SYNCONRETURN(NO) LANGUAGE_BLOCK(283FE8CC) COMMAREA(00000000 , 00000000) LINK_LEVEL(2)
SYSEIB_REQUEST(NO)

TASK-00123 KE_NUM-006B TCB-QR /008E6968 RET-ASD1DE58 TIME-09:53:34.4935717033 INTERVAL-00.0601490158* -001111=
1-0000 01880000 000000DA 00000000 00000000 B81B4750 00000000 02000100 E3C8D4D7 *.....&.....*THMP*
:.....*
01A0 00000000 00000050 00000048 407093E0 00000000 2568215E *.....&.....*\.....*
2-0000 F0C3F461 C1D2C5C1 018600C4 00000000 C4C6C8C1 D7D3C9F1 00000000 26204500 *0C4/AKEA.F.D....DFHAPL11.....*
:.....*
02C0 00000000 7943A408 00000000 00000000 *.....*

```





Complete your session evaluations online at www.SHARE.org/Seattle-Eval

58

We once again see transaction THMD (task #123) regain control. Almost immediately it too “**Program-checks**” as indicated by entry number =001111=.


We then see CICS begin the process of issuing the “**DFHSR0622**” message and prepare to take an SVC dump with code **SR0622** as indicated by entry number =001116=.

Next is the trace point AP 0781 exception entry, which will have some key information in its full trace entry that we can use to potential resolve the error. However, entry number =001127= is the short trace entry which does not contain this expanded information.

We the see the corresponding full trace entry to short entry number =001111=. As mentioned on an earlier slide, data area 1- is the parameter list and not fully documented. Data area 2- is the Kernel Error Data table entry, which we will again cover on the full trace counterpart to entry number =001127=.


Transaction Profile (THMD and THME) with Trace Entries

A look at asynchronous tasks plus a couple of abends (continued):
 AP 0781 SRP *EXC* - ABEND_ASRA PROGRAM(THMPGMC) OFFSET(00001E7C) EXEC_KEY(USER) OC4_ON(ECDSA)



```

TASK-00123 KE_NUM-006B TCB-QR /008E6968 RET-AS8B70C TIME-09:53:45.9127418593 INTERVAL-00.0000077187 =001127=
1-0000 E3C8D4D7 C7D4C340 F0C3F461 C1D2C5C1 00001E7C 00020781 *THMPGMC OC4/AKEA...@...a *
2-0000 02 *
3-0000 F0C3F461 C1D2C5C1 018600C4 00000000 C4C6C8C1 D7D3C9F1 00000000 26204500 *OC4/AKEA.f.D...DFHAPLIL.....*
0020 00000000 25A45700 00000000 26E47800 35A2C080 0000001E 00000004 FFFFFFFF *.....U...5{.....U.....*
0040 07952000 80000000 00000000 2932668C 00020004 2943A408 00000000 00000000 *.n.....U.....*
0060 90800000 00000000 00000000 27A173F8 00000000 27A161D0 00000000 2943A008 *.....8...../*
0080 00000000 00000000 00000000 2943A080 00000000 26E55798 00000000 26E55750 *.....V.q.....V.&*
00A0 00000000 5C000000 00000000 27A1A5A0 00000000 27A17298 00000000 29324950 *.....Z.....*
00C0 00000000 29325E74 00000000 2932490C 00000000 27A160A8 00000000 A932667A *.....Y.....Z...!*
00E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0100 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0120 00000000 01000002 07952000 80000000 00000000 2932668C 08020004 2943A408 *.....n.....U.....*
0140 00000000 00000000 90800000 00000000 00000000 27A173F8 00000000 27A161D0 *.....8...../*
0160 00000000 2943A008 00000000 00000000 00000000 2943A008 00000000 26E55798 *.....V.q.....*
0180 00000000 26E55750 00000000 5C000000 00000000 27A1A5A0 00000000 27A17298 *.....V.&.....*V.....q*
01A0 00000000 29324950 00000000 29325E74 00000000 2932490C 00000000 27A160A8 *.....Z.....*
01C0 00000000 A932667A 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
01E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0200 00000000 00000000 00000000 01000002 CEF0008 E1E95401 00000000 00000000 *.....7.Q.Z.....*
0220 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0240 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0260 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0280 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
02A0 45830E10 00000001 00010004 C0000000 00000000 27A1748E 25843200 0000123C *.....C.....{.....d.....*
02C0 00000000 2943A408 00000000 00000000 *.....U.....*
  
```



Complete your session evaluations online at www.SHARE.org/Seattle-Eval

59

This slide contains the full trace entry counterpart to short entry number =001127=. Once again this entry is documented in the CICS Trace Entries manual. First have a quick look at the attributes provided, **PROGRAM(THMPGMC)**, **OFFSET(00001E7C)**, and **OC4_ON(ECDSA)**. From these we gain knowledge of the offending program, plus the offset to the instruction within the program. We can also see that the problem is in accessing a storage area in the ECDSA (CICS mode storage) which we do not have access to, as indicated by the OC4_ON(ECDSA) attribute.


Data area 1- is labeled as **TCASRDAT** for trace point AP 0781 in the CICS Trace Entries manual. This provides us with the program name, KE error code, program offset, and some flags. These are what we see as the interpreted attributes mentioned above. Data area 2- is labeled as TCAPCSTG or the storage hit indicator. Data area 3- is the Kernel Error Data table entry. This is fully documented in the CICS Data Areas manual. However a few key offsets are:

- +0 – which is the System and User abend codes
- +40 – which is the PSW at the time of the abend
- +68 – which is the saved registers (8 bytes each) at the time of the abend

Since THMD transaction and its programs are defined with TASKDATAKEY(USER) and EXECKEY(USER) respectively we are not permitted to modify CICS mode storage. This is also an indication that we have specified SI override parameter **STGPROT(YES)**.

Transaction Profile (THMD and THME) with Trace Entries

A look at asynchronous tasks plus a couple of abends (continued):




00124	QR	RM	0202	RMUW	EXIT	BACKOUT_UOW/OK	RET-A5658C20	09:53:46.2982243750	00.0000248906	=002718=
XM	QR	PG	0801	PGXM	ENTRY	TERMINATE_TRANSACTION	RET-A565996C	09:53:46.2983048437	00.0000804687	=002719=
XM	QR	SM	0F01	SMAR	ENTRY	RELEASE_TRANSACTION_STG	RET-A56599AC	09:53:46.2983905000	00.0000108593	=002723=
XM	QR	XM	1306	XMTA	EXIT	TASK_REPLY/OK	RET-A562073C	09:53:46.3027302968	00.0001735937	=002741=
XM	QR	DS	0013	DSKE	EXIT	TASK_REPLY/OK	RET-A5604230	09:53:46.3027519218	00.0000216250	=002742=

**THME
#124**

00123	QR	RM	0202	RMUW	EXIT	BACKOUT_UOW/OK	RET-A5658C20	09:53:47.0138108283	00.0001950937	=004174=
XM	QR	PG	0801	PGXM	ENTRY	TERMINATE_TRANSACTION	RET-A565996C	09:53:47.0138956564	00.0000848281	=004175=
XM	QR	SM	0F01	SMAR	ENTRY	RELEASE_TRANSACTION_STG	RET-A56599AC	09:53:47.0139794533	00.0000107031	=004179=
XM	QR	XM	1306	XMTA	EXIT	TASK_REPLY/OK	RET-A562073C	09:53:47.0149345627	00.0001267343	=004199=
XM	QR	DS	0013	DSKE	EXIT	TASK_REPLY/OK	RET-A5604230	09:53:47.0149445627	00.0000100000	=004200=

**THMD
#123**

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



60

On the last slide of this series we have included the trace entries which show the end of both THMD and THME transactions. This to show that even if tasks do abend they must go through back-out processing as indicated by entry numbers =002718= and =004174=.

You will also notice that the storage for each associated task must be cleaned up as indicated by entry numbers b and =004179=. Even though the tasks have abended, another abend could occur at this point if one of the storage areas was involved in a storage violation.

Finally you will notice the corresponding "EXIT" trace entries to the "ENTRY" trace entries we covered at the start of both tasks. Trace entries =002741= and =002742= for task #124 (THME), plus entries =004199= and =004200= for task #123 (THMD) will indicate the true end for each task.


While both of these problems are uncovered due to SI parameter options we have chosen, please notice that CICS does not indicate them via any special message. Both simply appear as "Protection Exceptions" or OC4s. The real indication of the special CICS handling is indicated by the DFHSR0622 message and the dump with a code of SR0622. You should never suppress the SR0622 SVD dump since it may be the only way to resolve the problem.

- The next few slides represent our transactions THMH and THMI, which provides a look potential file control problems due to concurrent task execution.
- This not meant to be an in depth look at potential problems, but ones which might occur due to operating in the same region.

Transaction Profile (THMH and THMI) with Trace Entries

A look at tasks competing for file control resources (continued):

THMH
#91



```

00091 QR  AP E160 EXEC  ENTRY READ FILE('THMFLK1 ' AT X'27A0A828') INTO( AT X'27A0FCA4') LENGTH(256 AT X'27A0FA4F') RIDFLD( AT
X'A7A0F9EC') EQUAL UPDATE NOHANDLE COBOLII STMT_#(00375)
RET-8008661E 14:42:51.8331564846 00.0004177968 =000958=

00091 QR  DS 0004 DSSR  ENTRY WAIT_MVS RESOURCE_TYPE(FCIOWAIT) ECB_ADDRESS(275184E0) PURGEABLE(NO) WLM_WAIT_TYPE(IO)
RESOURCE_NAME(THMFLK1)
RET-A68914A4 14:42:51.8354433125 00.0014659060 =000964=

00092 QR  LD 0001 LDLD  ENTRY ACQUIRE_PROGRAM PROGRAM_TOKEN(00000048_40AC97F8)
RET-ASD35A9E 14:42:51.8368143750 00.0012942031 =000966=

00092 QR  LD 0002 LDLD  EXIT ACQUIRE_PROGRAM/OK ENTRY_POINT(A921D000) LOAD_POINT(2921D000) PROGRAM_LENGTH(9858)
PROGRAM_ATTRIBUTE(REUSABLE) LOCATION(ESDSA) COPY_STATUS(OLD_COPY) FETCH_TIME() LIBRARY
(DFHRPL) BUNDLE_INSTALLED_LIB(NO) PRIVATE_LIBRARY(NO)
RET-ASD35A9E 14:42:51.8368448906 00.0000305156 =000967=



THMI  
#92



```


00092 QR AP 00E1 EIP ENTRY GET-CONTAINER
REQ(0004) FIELD-A(27E0A698 .\wq) FIELD-B(09003414)
BOUNDARY(0200)
RET-A9222736 14:42:51.8421614221 00.0002841562 =001049=
00092 QR AP E160 EXEC ENTRY GET CONTAINER('THEMISCONT ' AT X'27E0FA3B') CHANNEL('THEMISCHNL ' AT X'A7E0FA2B') INTO(
AT X'27E102A4') FLENGTH(32 AT X'27E0FA4B') COBOLII STMT_#(00369)
RET-8008661E 14:42:51.8421976408 00.0000362187 =001050=

00092 QR AP 00E1 EIP EXIT GET-CONTAINER OK
REQ(00F4) FIELD-A(00000000) FIELD-B(00003414)
BOUNDARY(0200)
RET-A9222736 14:42:51.8427045471 00.0000078750 =001058=
00092 QR AP 00E1 EIP ENTRY READ
REQ(0004) FIELD-A(27E0A698 .\wq) FIELD-B(09000602)
BOUNDARY(0200)
RET-A922335D 14:42:51.8428161877 00.0001116406 =001059=
00092 QR AP E160 EXEC ENTRY READ FILE('THMFLK2 ' AT X'27E0A828') INTO(AT X'27E0FA44') LENGTH(256 AT X'27E0FA4F') RIDFLD(AT
X'A7E0F9F8') EQUAL UPDATE NOHANDLE COBOLII STMT_#(00379)
RET-8008661E 14:42:51.8428522814 00.0000360937 =001060=

00092 QR DS 0004 DSSR ENTRY WAIT_MVS RESOURCE_TYPE(FCIOWAIT) ECB_ADDRESS(27518760) PURGEABLE(NO) WLM_WAIT_TYPE(IO)
RESOURCE_NAME(THMFLK2)
RET-A68914A4 14:42:51.8436896564 00.0006376875 =001066=

00091 QR NQ 0301 NQED ENTRY ENQUEUE POOL_TOKEN(2756F500) ENQUEUE_NAME1(275186B8 , 00000004) ENQUEUE_NAME2(275186BC , 00000006)
WAIT(NO) - ARGUMENT(. .K.111111)
RET-A6887002 14:42:51.8437966721 00.0000458593 =001068=

THMH
#91



Complete your session evaluations online at www.SHARE.org/Seattle-Eval


```


```

In entry number =000958= we see the level 2 EI information. The **READ FILE() UPDATE** request is for file **THMFLK1** which will read a **256 byte** record “**INTO**” a **working storage area**. The request results in an I/O being necessary as evidenced by entry number =000964=, which is a DS domain **WAIT_MVS** with a **RESOURCE_TYPE(FCIOWAIT)**.

We then see task #92 regain control to acquire access to program **THMPGMB**. No actual load is required since task #91 paid the price to load the program into memory.


Since this task is using the same program, we see the **GET CONTAINER** command being performed in entry numbers =001049=, =001050=, and =001058=. While the container and channel names, **THEMISCONT** and **THEMISCHNL** respectively, appear to be the same you should know that each task will have its own unique channel and set of containers.

Next we see the **READ FILE() UPDATE** request for file **THMFLK2** being performed as seen in entry numbers =001059= and =001060=. This request is also for a **256 byte** record to be read “**INTO**” a **working storage area**. Once again the request results in an I/O being issued as seen in entry number =001066=, which is a DS domain **WAIT_MVS** with a **RESOURCE_TYPE(FCIOWAIT)**.

Next we see task #91 regain control after its I/O request completes and CICS performs an ENQ command, entry number =001068= for a key of **111111**. CICS will always do this enqueue for the key of the record plus the address of the base cluster control block or **BCCB**. This is the way CICS protects two tasks from updating the same record.

Transaction Profile (THMH and THMI) with Trace Entries

A look at tasks competing for file control resources (continued):




```

00091 QR  NQ 0302 NQED EXIT ENQUEUE/OK RET=A6887002 14:42:51.8438679533 00.0000460312 =001071=
00091 QR  AP E161 EXEC EXIT READ FILE('THMFLK1 ' AT X'27A0A828') INTO(' 111111C. BAKER OTTAWA, ONTARIO
5121200326 11 8150011.00***** AT X'27A0FA4') LENGTH(80 AT X'27A0FA4') RIDFLD( AT
X'A7A0F9EC') EQUAL UPDATE RESP(0) RESP2(0) NOHANDLE COBOLII STMT_#(00375) RET=8008889C 14:42:51.8440596252 00.0000445000 =001081=
00091 QR  AP 00E1 EIP EXIT READ OK REQ(00F4) FIELD-A(00000000 ....) FIELD-B(00000602 ....)
BOUNDARY(0200) RET=A9222ED4 14:42:51.8440665939 00.0000069687 =001082=
00092 QR  NQ 0301 NQED ENTRY ENQUEUE POOL_TOKEN(2756F500) ENQUEUE_NAME1(27518938 , 00000004) ENQUEUE_NAME2(2751893C , 00000006)
WAIT(NO) - ARGUMENT(..Ly999999) RET=A6887002 14:42:51.8531777187 00.0000660625 =001191=
00092 QR  NQ 0302 NQED EXIT ENQUEUE/OK RET=A6887002 14:42:51.8532072031 00.0000111562 =001194=
00092 QR  AP E161 EXEC EXIT READ FILE('THMFLK2 ' AT X'27E0A828') INTO(' 9999999ANJII STEVENS RALEIGH, N.Y.
8459163926 11 8150049.00***** AT X'27E0FA4') LENGTH(80 AT X'27E0FA4') RIDFLD( AT
X'A7E0F9F8') EQUAL UPDATE RESP(0) RESP2(0) NOHANDLE COBOLII STMT_#(00379) RET=8008889C 14:42:51.8544258750 00.0000508281 =001206=
00092 QR  AP 00E1 EIP EXIT READ OK REQ(00F4) FIELD-A(00000000 ....) FIELD-B(00000602 ....)
BOUNDARY(0200) RET=A9223350 14:42:51.8544337656 00.0000078906 =001207=
00091 QR  AP 00E1 EIP ENTRY READ REQ(0004) FIELD-A(27A0A698 ..wq) FIELD-B(09000602 ....)
BOUNDARY(0200) RET=A9222FC8 14:42:54.0518932658 00.0000202656 =001578=
00091 QR  AP E160 EXEC ENTRY READ FILE('THMFLK2 ' AT X'27A0A828') INTO( AT X'27A0FDA4') LENGTH(256 AT X'27A0FA4') RIDFLD( AT
X'A7A0F9F2') EQUAL UPDATE NOHANDLE COBOLII STMT_#(00375) RET=8008661E 14:42:54.0525958283 00.0007025625 =001579=
00091 QR  NQ 0301 NQED ENTRY ENQUEUE POOL_TOKEN(2756F500) ENQUEUE_NAME1(27518E38 , 00000004) ENQUEUE_NAME2(27518E3C , 00000006)
WAIT(NO) - ARGUMENT(..Ly999999) RET=A6887002 14:42:54.0530352658 00.0000400781 =001593=
00091 QR  NQ 0302 NQED EXIT ENQUEUE/EXCEPTION REASON(ENQUEUE_BUSY) RET=A6887002 14:42:54.0530417658 00.0000065000 =001594=
00091 QR  DS 0004 DSSR ENTRY SUSPEND SUSPEND_TOKEN(06080005) RESOURCE_TYPE(ENQUEUE) PURGEABLE(YES) DEADLOCK_ACTION(Delayed)
WLM_WAIT_TYPE(LOCK) RESOURCE_NAME(FCDSRECD) RET=A5BE69B6 14:42:54.0530568908 00.0000098593 =001596=

```

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



64

Task #91 was able to acquire the ENQ lock as evidenced by trace entry number =001071=.

We then see the READ UPDATE command complete as seen in entry numbers =001081= and =001082=. In fact the INTO() attribute contains the data record or at least some of it. Also the command completed successfully as indicated by the RESP() and RESP2() attributes. After this command, task #91 performs a function which causes it relinquish control back to CICS.

We then see task #92 regain control after its I/O completes. Again this is evidenced by the successful ENQ command issued by CICS in entry numbers =001191= and =001192=. The enqueue is for the BCCB of THMFLK2 with a key of 999999.


With the lock on the record we then see the completion of the READ UPDATE request in entry numbers =001206= and =001207=. Once again you can see a portion of the record in the INTO() attribute. The command does complete NORMAL, as indicated by the RESP() and RESP2() attributes. This task then performs a function which relinquishes control back to CICS.

Task #91 regains control and performs a second READ FILE() UPDATE as seen in entry numbers =001578= and =001579=. This is for file THMFLK2 to read a 256 byte record into a working storage area.


The second READ FILE() cannot proceed since the ENQ request for the key of 999999 against the BCCB of file THMFLK2 fails with a exception REASON(ENQUEUE_BUSY). Task #92 has already read the same record and obtained the lock on it. This can be seen in entry numbers =001593= and =001594=. The task now enters a wait state as seen in entry number =001596=. The RESOURCE_TYPE() and RESOURCE_NAME() indicate the actual wait condition.

Transaction Profile (THMH and THMI) with Trace Entries


A look at tasks competing for file control resources (continued):




00092 QR	AP 00E1 EIP	ENTRY	READ	BOUNDARY(0200)	REQ(0004) FIELD=A(27E0A698 \wq) FIELD=B(09000602 . . .)	RET-A9223444 14:42:55.0996131562 00.0024538278 =002318=
00092 QR	AP E160 EXEC	ENTRY	READ FILE('THMFLK1 ' AT X'27E0A828')	INTO(AT X'27E0FA4')	LENGTH(256 AT X'27E0FA4')	RIDFLD(AT X'A7E0F9FE') EQUAL UPDATE NOHANDLE COBOLII STMT_#(00380)
						RET-8008661E 14:42:55.0996936250 00.0000804687 =002319=
00092 QR	DS 0004 DSSR	ENTRY	SUSPEND SUSPEND_TOKEN(060A0005)	RESOURCE_TYPE(FCPSSUSP)	PURGEABLE(YES) WLM_WAIT_TYPE(TO)	RET-A688C32A 14:42:55.0999271406 00.0000315468 =002329=
				RESOURCE_NAME(THMFLK1)		
00091 QR	DS 0005 DSSR	EXIT	SUSPEND/PURGED REASON(TIMED_OUT)	COMPLETION_CODE(N)		RET-A58E69B6 14:43:09.5146541093 00.5201538278 =002931=
00091 QR	DS 0004 DSSR	ENTRY	RESUME SUSPEND_TOKEN(06080005)	COMPLETION_CODE(N)		RET-A58E6EE2 14:43:09.5146816718 00.0000275625 =002932=
00091 QR	DS 0005 DSSR	EXIT	RESUME/EXCEPTION REASON(TIMED_OUT)			RET-A58E6EE2 14:43:09.5146897812 00.0000081093 =002933=
00091 QR	NQ 0302 NQED	EXIT	ENQUEUE/PURGED REASON(TIMED_OUT)			RET-A6886D00 14:43:09.5147007998 00.0000110156 =002934=
00091 QR	AP 04B6 FCVS	"EXC"	PURGED			RET-A668FDE8 14:43:09.5147134062 00.0000126093 =002935=
00091 QR	ME 0502 MEIN	EXIT	INQUIRE_MESSAGE_DATA/OK GLOBALS_PTR(00000048_40200484)	MSG_PTR(00000048_40203089)	DEST_PTR(00000048_402030D0)	EXIT_PTR(00000048_40203176)
			ROUTE_CODES_BUFFER(26E7BD4D , 00000000 , 0000001C)	QUEUES_BUFFER(26E7BC7 , 00000001 , 00000019)	ACTION(C) SEVERITY(C) NORDER(OUTE(0)	TIME(144309) DATE(30012015) DATE_FORMAT(M) SYSID(TCA3) SPECIFIC_APPLID(CICSA3)
			TRANSACTION_ID(THMH) TERMINAL_ID(????) PROGRAM_NAME(THMPGMB) USERID(CICSUSER) NETNAME(?????????)	PRIMARY_ABCODE(C) SECONDARY_ABCODE(AFCY)		RET-25680B67 14:43:09.9344223125 00.0000148437 =005437=
00092 QR	DS 0004 DSSR	ENTRY	RESUME SUSPEND_TOKEN(06080005)			RET-A6889892 14:43:15.3026850625 00.0000337500 =006010=
00092 QR	DS 0005 DSSR	EXIT	RESUME/OK			RET-A6889892 14:43:15.3026936718 00.0000086093 =006011=
00092 QR	NQ 0301 NQED	ENTRY	ENQUEUE POOL_TOKEN(2756F500)	ENQUEUE_NAME1(2818BBE8 , 00000004)	ENQUEUE_NAME2(2818BBE8 , 00000006)	RET-A6887002 14:43:15.3027720156 00.0000783437 =006012=
			WAIT(NO) - ARGUMENT(. . .K.111111)			RET-A6887002 14:43:15.3027830937 00.0000110781 =006013=
00092 QR	NQ 0302 NQED	EXIT	ENQUEUE/EXCEPTION REASON(ENQUEUE_BUSY)			RET-A6886D00 14:43:15.3027910625 00.0000079687 =006014=
00092 QR	NQ 0301 NQED	ENTRY	ENQUEUE POOL_TOKEN(2756F500)	ENQUEUE_NAME1(2818BBE8 , 0000000A)	- ARGUMENT(. . .K.111111)	



THMI
#92




THMH
#91



THMI
#92

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



65

Task #92 now regains control and performs its second READ FILE() UPDATE for file THMFLK1 as seen in entry numbers =002318= and =002319=. This request ends in a wait state as seen in entry number =002329=. The DS domain places the task into a wait with a RESOURCE_TYPE(FCPSSUSP) and RESOURCE_NAME(THMFLK1). This indicates a wait for a private string.


Next we see task #91 regain control as evidenced by entry numbers =002931=, =002932=, and =002933=. However, this is not a good event since we see an exception REASON(TIMED_OUT).

Entry number =002943= indicates that the ENQ request has failed to obtain the lock as task #92 still controls the record. This is a direct result of the fact that the transaction was defined with the SPURGE(YES) attribute with a reasonable value for DTIMOUT(). Without these attributes the task would have stayed in the wait state indefinitely and potentially cause CICS to "stall" as more and more tasks get backed up.

We then see the task get purged and the transaction dump of AFCY is produced, as evidenced by entry numbers =002935= and =005437=. The many trace entries eliminated between these two were CICS producing the transaction dump.


Task #92 now regains control since a string has been release with the purging of task #91. This can be seen in entry numbers =006010= and =006011=. However, task #91 has not completed its cleanup so this task cannot obtain the ENQ lock for the BCCB of THMFLK1 with a key of 111111 as seen in entry numbers =006012=, =006013=, and =006014=. Since task #91 still holds the ENQ this task receives the REASON(ENQUEUE_BUSY) status.

Transaction Profile (THMH and THMI) with Trace Entries



A look at tasks competing for file control resources (continued):

Task	QR	DS	0004	DSSK	ENTRY	SUSPEND	SUSPEND_TOKEN(000A0005)	RESOURCE_TYPE(ENQUEUE)	PURGEABLE(YES)	DEADLOCK_ACTION(Delayed)	WLM_WAIT_TYPE(LOCK)	RESOURCE_NAME(FCDSRECD)	Response	
THMI #92	00092	QR	DS	0004	DSSK	ENTRY	SUSPEND	SUSPEND_TOKEN(000A0005)	RESOURCE_TYPE(ENQUEUE)	PURGEABLE(YES)	DEADLOCK_ACTION(Delayed)	WLM_WAIT_TYPE(LOCK)	RESOURCE_NAME(FCDSRECD)	RET-A58E69B6 14:43:15.3028250000 00.0000339375 =006015=
	00091	QR	RM	0202	RMUW	EXIT	BACKOUT_UOW/OK							RET-A5658c20 14:43:15.3346995471 00.0005363437 =006205=
	XM	QR	SM	0F01	SMAR	ENTRY	RELEASE_TRANSACTION_STG							RET-A56599AC 14:43:15.3349385783 00.0000113906 =006210=
	XM	QR	XM	1306	XMTA	EXIT	TASK_REPLY/OK							RET-A562073C 14:43:15.3356345939 00.0001049531 =006224=
	XM	QR	DS	0013	DSKE	EXIT	TASK_REPLY/OK							RET-A5604230 14:43:15.3356444377 00.0000098437 =006225=
THMH #91	00092	QR	NQ	0302	NQED	EXIT	ENQUEUE/OK							RET-A6886D00 14:43:15.3357020471 00.0000121875 =006227=
	00092	QR	AP	E161	EXEC	EXIT	READ FILE('THMFLK1 ' AT X'27E0A828') INTO(' 111111C. BAKER OTTAWA, ONTARIO 5121200326 11 8150011.00***** AT X'27E0FFA4') LENGTH(80 AT X'27E0FA4F') RIDFLD(AT X'A7E0F9FE') EQUAL UPDATE RESP(0) RESP2(0) NOHANDLE COBOLII STMT_#(00380)							RET-8008889C 14:43:15.3384742189 00.0002186562 =006239=
	00092	QR	AP	00E1	EIP	EXIT	READ OK BOUNDARY(0200)							REQ(00F4) FIELD-A(00000000 ...) FIELD-B(00000602 ...) RET-A9223444 14:43:15.3384902971 00.0000160781 =006240=
	00092	QR	RM	FA12	RMUO	EXIT	COMMIT_UOW/OK FAILED_LINK(00000000)							RET-A5659640 14:43:15.3465475312 00.0000220156 =006318=
	XM	QR	SM	0F01	SMAR	ENTRY	RELEASE_TRANSACTION_STG							RET-A56599AC 14:43:15.3467724062 00.0000086875 =006323=
	XM	QR	XM	1306	XMTA	EXIT	TASK_REPLY/OK							RET-A562073C 14:43:15.3473221093 00.0001701718 =006336=
	XM	QR	DS	0013	DSKE	EXIT	TASK_REPLY/OK							RET-A5604230 14:43:15.3473331093 00.0000110000 =006337=
THMI #92														



Complete your session evaluations online at www.SHARE.org/Seattle-Eval

66

Task #92 now enters a wait state as seen in entry number =006015=. DS domain indicates the wait condition with a **RESOURCE_TYPE(ENQUEUE)** and a **RESOURCE_NAME(FCDSRECD)**.

Finally task #91 completes its cleanup of resources as a result of being purged as seen by entry numbers =006205=, =006210=, =006224=, and =006225=. These represent the events which occur at task end.


Task #92 regains control with the obtaining of the ENQ lock as seen in entry number =006227=.

We then see the completion of the READ FILE() request for file THMFLK1 as seen in entry numbers =006239= and =006240=. We can see the first portion of the record in the **INTO()** attribute, plus the indication of **NORMAL** completion by the **RESP()** and **RESP2()** attributes.

Task #92 then proceeds with the rest of its function and ends normally as seen in entry numbers =006318=, =006323=, =006336=, and =006337=.

- The next few slides represent our transaction MF00, which invokes a web service to acquire some data from a provider application.
- This not meant to be an in depth look at all CICS web service process, just a quick overview.

Transaction Profile (MF00) with Trace Entries



SHARE
Secure. Reliable. Visible.

A quick look at a CICS web service (continued):

```

00136 QR  AP 00E1 EIP  ENTRY PUT-CONTAINER                REQ(0004) FIELD-A(27F09FF8 .0.8) FIELD-B(09003416 ...)
                                BOUNDARY(0200)                RET-A7515784 08:59:06.6672987033 00.0000201718 =000615=
00136 QR  AP E160 EXEC ENTRY PUT CONTAINER('DFHWS-DATA      ' AT X'27F0BDC0') CHANNEL('THMF00-WEBSCH  ' AT X'A7F0BDD0') FROM(
                                AT X'27F0C000') FLENGTH(84 AT X'27F0BFF8') COBOLII STMT_#(00044)
                                RET-8008661E 08:59:06.6673276096 00.0000289062 =000616=

00136 QR  AP 00E1 EIP  EXIT  PUT-CONTAINER OK           REQ(00F4) FIELD-A(00000000 ...) FIELD-B(00003416 ...)
                                BOUNDARY(0200)                RET-A7515784 08:59:06.6680786408 00.0000162656 =000626=

00136 QR  AP 00E1 ETP  ENTRY INVOKE-SERVICE             REQ(0004) FIELD-A(27F09FF8 .0.8) FIELD-B(0900C002 ...)
                                BOUNDARY(0200)                RET-A75157E2 08:59:06.6680942814 00.0000156406 =000627=
00136 QR  AP E160 EXEC ENTRY INVOKE SERVICE('THMF00Req
                                X'27F0BDD0') URIMAP('THMF00UR' AT X'A7F0BEF0') OPERATION('THMFW00operation
                                ' AT X'27F0BDD0') COBOLII
                                STMT_#(00045)                RET-8008661E 08:59:06.6683496096 00.0002553281 =000628=

00136 L800D SO 0201 SOCK ENTRY SEND SOCKET_TOKEN(01020000) TIMEOUT_VALUE(FFFFFFFF) BUFFER_LIST(26bc097c , 00000002)
                                RET-A6074FF6 08:59:06.7317824064 00.0012882189 =000906=
00136 L800D SO 0009 SOSO ENTRY SEND
                                RET-A5F8A17E 08:59:06.7323552971 00.0005728906 =000907=


00137 QR  SO 000C SOSO EXIT  RECEIVE
                                RET-A5F89A4E 08:59:06.7867856721 00.0002762187 =001115=
00137 QR  SO 0202 SOCK EXIT  RECEIVE/OK RECEIVE_BUFFER(27518000 , 00000316 , 00001000)
                                RET-A605F472 08:59:06.7868019846 00.0000163125 =001116=

00136 L800E SO 0201 SOCK ENTRY SEND BUFFER_LIST(26E84624 , 00000002)
                                RET-A60560E2 08:59:06.8846260156 00.0000852812 =001663=
00136 L800E SO 0009 SOSO ENTRY SEND
                                RET-A5F8A17E 08:59:06.8852597968 00.0006337812 =001664=

00136 QR  AP E161 EXEC EXIT  INVOKE SERVICE('THMF00Req
                                X'27F0BDD0') URIMAP('THMF00UR' AT X'A7F0BEF0') OPERATION('THMFW00operation
                                ' AT X'27F0BDD0') CHANNEL('THMF00-WEBSCH  ' AT
                                ' AT X'27F0BDD0') RESP(0)
                                RET-8008889C 08:59:06.9291608283 00.0001930312 =002025=
00136 QR  AP 00E1 EIP  EXIT  INVOKE-SERVICE OK           REQ(00F4) FIELD-A(00000000 ...) FIELD-B(0000C002 ...)
                                BOUNDARY(0200)                RET-A75157E2 08:59:06.9291696096 00.0000087812 =002026=

```

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



SHARE
in Seattle 2015

68

Task #136 is the “requester” application in our case. So we simply show the PUT CONTAINER being executed first, entry numbers =000615=, =000616=, and =000626=. You will notice that the CONTAINER name is DFHWS-DATA. This is the default container name used to hold your data, both request and response, unless you specify the CONTID option in DFHLS2WS or DFHWS2LS jobs.

Next we see the INVOKE SERVICE command being performed, entry numbers =000627= and =000628=. The SERVICE name, THMF00Req, is the name for this CICS requester application. We used a URIMAP, named THMF00UR, to supply attributes like host name, server location, listener port, and path string. This avoids specifying this information as part of the INVOKE SERVICE command, plus it makes it easier to change this information as necessary. The OPERATION is required and it reflects the provider application program name, THMFW00, suffixed by the word “Operation” in mixed case as you see it.

Next we see the SEND of the REQUEST message being performed, entry numbers =000906= and =000907=. We chose to show the “ENTRY” trace entries since it will show the actual message buffers in the full trace entry, shown next.


Task #137 is the CICS CWXN transaction executing to process the inbound request and invoke the provider application. We chose to only show the RECEIVE command being done, entry numbers =001115= and =001116=. Here we chose to show the “EXIT” trace entries, entry numbers =001115= and =001116=, since these will also show the message buffers, shown later.

Task #138 is the CPIH transaction executing which will run our provider application program, THMFW00. Here we chose to show the SEND of the RESPONSE, entry numbers =001663= and =001664=. Once again showing the “ENTRY” trace entries which will contain the message buffers.

Back to task #136 to simply show the “EXIT” entries for the INVOKE SERVICE command being done.

Transaction Profile (MF00) with Trace Entries

A quick look at a CICS web service (continued):
SO 0201 SOCK ENTRY - FUNCTION(SEND) SOCKET_TOKEN(01020000) TIMEOUT_VALUE(FFFFFFFF) BUFFER_LIST(260C097C , 00000002)




```

TASK-00136 KE_NUM-006A TCB-L800D/008B9988 RET-A6074FF6 TIME-08:59:06.7317824064 INTERVAL-00.0012882189 =000906=
1-0000 01300000 00000198 00000000 00000000 80000804 00020000 01000000 00000000 *.....*
0120 260C097C 00000002 00000000 00000000 *..@.....*
2-0040 276F2A0D 0000011E 00100738 000001F8 *?.....8*
3-0000 504F5354 202F6375 73744061 696E742F 54484046 30305072 6F762048 5454502F *POST /custMaint/THMF00Prov HTTP/*
0020 312E310D 0A486F73 743A206C 6F63161C 686F7374 3A393038 33000A53 4F415041 *1.1..Host: localhost:9083..SOAP*
0040 3274606F 6F3A2027 270D0A55 7365722B 4167656E 743A2049 424D5F43 4943535F *action: ""..User-Agent: THM_circ_*
0060 542616E 73616374 696F6E5F 53657276 65722F35 2E322E30 287A4F53 290D0A43 *Transaction_Server/5.2.0(c20S)..C*
0080 6F63465 6E742D54 7970653A 20746578 742F786D 6C3B2063 68617273 65743D55 *Content-Type: text/xml; charset=U*
00A0 5446328 000A436F 6E74656E 742D4C65 6E677468 3A203530 340D0A43 6F6E6E65 *TF-8..Content-Length: 504..Conne*
00C0 6374695F 6E3A2048 6565702B 416C6976 650D0A44 6174653A 204D6F6E 2C2D3032 *ction: keep-alive..Date: Mon, 02*
00E0 2046656E 20323031 35203133 3A35393A 30362047 4D542020 0D0A5445 3A207472 * Feb 2013 13:59:06 GMT ..TE: tr*
0100 61696C65 273000A 41636365 70742D45 6E636F64 696E673A 20200D0A 0D0A *ailers..Accept-Encoding: ....*
4-0000 3C534F41 50D0454E 563A456E 76656C6F 70652078 6D6C6E73 3A534F41 502D454E *SOAP-ENV:Envelope xmlns:SOAP-EN*
0020 563D2268 7474703A 2F2F7363 68656D61 732E786D 6C736F61 702E6F72 672F736F *v="http://schemas.xmlsoap.org/so*
0040 61702F65 6E78656C 6F70652F 22203E3C 534F4150 20454E56 3A426F64 793E3C54 *ap/envelope/">SOAP-ENV:Body<T*
0060 484D4657 3030407D 65726174 696F6E20 786D6C6E 73302268 7474703A 2F2F7777 *THMF00operation xmlns="http://ww*
0080 772E5448 40465733 302E5745 4246494C 452E5265 71756573 742E636F 6D223E3C *w.THMF00.WEBFILE.Request.com"<*
00A0 7673616D 5F726563 6F72645F 61726561 3E3C666C 5F657272 6F725F63 6F64653E *vsam_record_area<fl_error_code>*
00C0 3C2F666C 5F657272 6F725F63 6F64653E 3C666C5F 66756E63 745F636F 64653E49 *</fl_error_code><fl_func_code>I*
00E0 4E513C2F 666C5F66 73E6374 5F636F64 653E3C66 6C5F7265 636F7264 3E3C666C *NQ</fl_func_code><fl_record>fl*
0100 5F726865 793E3030 303E3130 30303C2F 666C5F72 6865793E 3C666C5F 636D706E *_rkey>00001000</fl_rkey><fl_cmpn*
0120 616D653E 3C2F666C 5F636F70 6E616D65 3E3C666C 5F616464 723E3C2F 666C5F61 *ame</fl_cmpname><fl_addr><fl_a*
0140 6464723E 3C666C5F 70686F6E 653E3C2F 666C5F70 686F6E65 3E3C666C 5F62616C *ddr><fl_phones></fl_phones><fl_bal*
0160 616E6365 3E3C2F66 6C5F6261 6C616E63 653E3C66 6C5F7261 74696E67 3E3C2F66 *ances</fl_balance><fl_rating></f*
0180 6C5F7261 74696E67 3E3C666C 6F636F6E 74616374 3E3C2F66 6C5F636F 6E746163 *l_rating</fl_contact></fl Contac*
01A0 743E3C2F *t></fl_record><vsam_record_area*
01C0 3E3C2F54 4 *"></THMF00operation></SOAP-ENV:B*
01E0 6F64793E 2 *ody></SOAP-ENV:Envelope> *

```

(1) Data areas are:

- 1 - Parameter list
- 2 - Address/Length pairs of buffers
- 3 - First buffer (max. 4000)
- 4 - second buffer (max. 4000)



Complete your session evaluations online at www.SHARE.org/Seattle-Eval

69

For task #136, here is the full trace entry =000906=. This is the send of the REQUEST message to the web service provider application. As you can see, its content could be useful at times. Data area 1- is the parameter list, not well documented. Data area 2- is a list of **buffer address/length** pairs. Data area 3- is the data at the first buffer address in the list, maximum of 4000 bytes shown. Data area 4- is the data at the second buffer address in the list.

The information in data area 3- is an **HTTP** header which provides key information about the request. If you look closely you will see the following:


- 1) "POST" – which is the HTTP method being performed
- 2) "/custMaint/THMF00Prov" – which is the path and provider web service name
- 3) "Host: localhost:9083" – which is the host name and listener port

Plus various other information. The format of this data area is document an HTTP manual and not in any CICS documentation.

The information in data area 4- is the actual SOAP envelope which is being sent to the provider location and eventually the provider application. As you can see this is a well-formed XML document which contains the tags and data that makes up the SOAP headers and body. The SOAP body will represent the structure you have defined as part of the DFHLS2WS job.

Transaction Profile (MF00) with Trace Entries


A quick look at a CICS web service (continued):
 SO 0202 SOCK EXIT - FUNCTION(RECEIVE) RESPONSE(OK) RECEIVE_BUFFER(27518000 , 00000316 , 00001000)



```


TASK-00137 KE_NUM=00C TCB=QR /008E6968 RET=A605F472 TIME=08:59:06.7868019846 INTERVAL=00.0000163125 =001116=
1-0000 01300000 00000198 00000000 00000000 83008002 02000000 03000100 00000000 *.....*
.....*
0120 00000000 00000000 00000000 00000000 *.....*
2-0000 504F5354 202F6375 73744061 696E742F 54484D46 30305072 6F762048 5454502F *POST /custMaint/THMF00Prov HTTP/*
0020 312E310D 0A486F73 743A206C 6F63616C 686F7374 3A393038 3300A53 4F415041 *1.1.Host: localhost:9083..SOAP*
0040 6374696F 6E3A2022 220D0A55 7365722D 4167656E 743A2049 42405F43 4943535F *ction: ""..User-Agent: IBM_CICS_*
0060 4726166F 73616374 696F6E5F 53657276 65727635 2E327E30 287A4F53 29000A43 *transaction_Server/5.2.0(200)..C*
0080 6A6E7465 6E742D54 7970653A 20746578 742F786D 6C3E2063 68617273 65743D55 *ontent-Type: text/xml; charset=U*
00A0 54742D38 0D0A436F 6E74656E 742D4C65 6E677468 3A203530 34000A43 6F6E6E65 *TF-8..Content-Length: 504..Conne*
00C0 6374696F 6E3A2048 6565702D 416C6976 65000A44 6174653A 2D406F6E 2C203032 *ction: Keep-Alive..Date: Mon, 02*
00E0 2046696E 20323031 35203133 3A35393A 30362047 4D542020 0D0A5445 3A207472 * Feb 2015 13:59:06 GMT ..TE: tr*
0100 61696C65 72730D0A 41636365 70742D45 6E636F64 696E673A 20200D0A 0D0A3C53 *ailers..Accept-Encoding: ....S*
0120 4F41502D 454E563A 456E7665 6C6F7065 20786D6C 6E733A53 4F41502D 454E563D *OAP-ENV:Envelope xmlns:SOAP-ENV=*
0140 22687474 703A2F2F 73636865 6D61732E 786D6C73 6F61702E 6F72672F 736F6170 *"http://schemas.xmlsoap.org/soap*
0160 2F656E76 6C6C6F70 652F2220 3E3C534F 41502D45 4E563A42 6F64793E 3C54484D */envelope/" ><SOAP-ENV:Body><TM*
0180 46573030 4F706572 6174696F 6E20786D 6C6E733D 22687474 703A2F2F 7777772E *Pw00operation xmlns="http://www.*
01A0 54484046 5730702E 57454246 494C452E 52657175 6573742E 636F6D22 3E3C7673 *THMF00.WEBFILE.Request.com"><vs*
01C0 616D5F72 65636E72 645F6172 65613E3C 666C5F65 72726F72 5F636F64 653E3C3F *am_record_area<fl_error_code></*
01E0 666C5F65 72726F72 5F636F64 653E3C6E 6C5F6675 6663745F 636F6465 3E494E51 *fl_error_code<fl_func_code>INQ*
0200 3C2F666C 5F66756A 63745F63 6F64653E 3C666C5F 7265636F 72643E3C 666C5F72 *</fl_func_code>fl_record<fl_r*
0220 6B65793E 30303030 61303030 3C2F666C 5F726B65 793E3C66 6C5F636D 706E616D *key=00001000<fl_rkey><fl_cmpnam*
0240 653E3C2F 666C5F63 61300E61 6D653E3C 666C5F61 6464723E 3C2F666C 5F616464 *e></fl_cmpname<fl_addr></fl_add*
0260 723E3C66 6C5F7068 6F6E653E 3C2F666C 5F70686F 6E653E3C 666C5F62 616E616E *r></fl_phone<fl_phone<fl_balanc*
0280 63653E3C 2F666C5F 626E6361 6E63653E 3C666C5F 72617469 6E673E3C 2F666C5F *ce></fl_balance<fl_rating></fl_*
02A0 72617469 6E673E3C 666C5F63 6F6E7461 63743E3C 2F666C5F 636F6E74 6163743E *rating<fl_contact><fl_contact>*
02C0 3C2F666C 5F726563 6F726563 3C2F7673 616D5F72 65636F72 645F6172 65613E3C *</fl_record></vsam_record_area>*
02E0 2F54484D 46573030 4F706572 6174696F 6E3E3C2F 534F4150 2D454E56 3A426F64 */THMF00operation</SOAP-ENV: Bod*
0300 793E3C2F 534F4150 2D454E56 6A456E76 656C6F70 653E *y></SOAP-ENV:Envelope>
  
```

(1) Data areas are:
 1 - Parameter list
 2 - Data Buffer



Complete your session evaluations online at www.SHARE.org/Seattle-Eval

Here we show task #137, the **CWXXN** transaction to initial the actual CICS provider task. This trace entry, number =**0001116**=, is the full trace entry for the RECEIVE being done on the CICS server specified in the URIMAP. Data area 1- is the parameter list, not well documented. Data area 2- is the message buffer of the request SOAP message plus the HTTP header. Here you will notice that the HTTP header and SOAP envelope are now contained in a single message buffer.



Transaction Profile (MF00) with Trace Entries

A quick look at a CICS web service (continued):
 SO 0201 SOCK ENTRY - FUNCTION(SEND) BUFFER_LIST(20E84624 , 00000002)


```

TASK-00138 KE_NUM=006F TCB=L800E/0088B2D8 RET=A6056DE2 TIME=08:59:06.8846260156 INTERVAL=00.0000852812 =001663=
1-0000 01300000 00000198 00000000 00000000 80000000 00060000 01000000 00000000 *.....*
:.....*
0120 26E84624 00000002 00000000 00000000 *..Y.....*
2-0000 26A65A40 000000E8 26A65800 0000023E *..w!...Y.w.....*
3-0000 48545450 2F312E31 20323030 204F4820 20202020 20202020 20202020 20202020 *HTTP/1.1 200 OK *
0020 20202020 20202020 20202020 20000A44 6174653A 204D6F6E 2C203032 20466562 * ..Date: Mon, 02 Feb*
0040 70232031 35201133 3A35393A 30362047 40542020 000A5365 72766577 3A204942 * 2015 11:50:06 GMT ..Server: TR*
0060 40544349 43535F54 72616E73 61637469 6F6E5F53 65727665 722F352E 322E3028 *M_CICS.Transaction_Server/5.2.0(*
0080 7A764329 0D0A436F 6E74656E 742D5479 70653A20 74657874 2F786D6C 3B206368 *205)..Content-Type: text/xml; ch*
00A0 61727865 74305554 462D3800 0A436F6E 74656E74 2D4C656E 6774683A 20303030 *charset=UTF-8..Content-Length: 000*
00C0 30303030 30303030 30353734 000A436F 6E6E6563 74696F6E 3A204865 65702D41 *000000000574..Connection: Keep-A*
00E0 6C69786E 000A0D0A *live... *
4-0000 3C534F41 02D454E 563A456E 76656C6F 70652078 606C6E73 3A534F41 502D454E *<SOAP-ENV:Envelope xmlns:SOAP-ENV*
0020 563D2268 7474703A 2F2F7363 68656061 732E786D 6C736F61 70266F72 672F736F *V="http://schemas.xmlsoap.org/so*
0040 61702F65 6E78656C 6F70652F 22203C3C 534F4150 2D454E56 3A426F64 793E3C54 *ap/envelope/" ><SOAP-ENV:Body><T*
0060 484D4657 30303070 65726174 606F6E52 6573706F 6E736520 786D6C6E 733D2268 *TMFW00OperationResponse xmlns="ht*
0080 7474703A 2F2F7363 772E5448 40465730 302E5745 4246494C 452E5265 73706F6E *ttp://www.TMFW00.WEBFILE.Respon*
00A0 73652863 6F6D2230 2C767363 605F7265 636F7264 5F617365 613E3C66 6C5F6572 *se.com"><vsam_record_area<fl_er*
00C0 726F725F 636F6465 3E303C2F 666C5F65 72726F72 5F636F64 653E3C66 6C5F6675 *ror_code=0</fl_error_code><fl_fu*
00E0 6E63745F 636F6465 3E494E51 3C2F666C 5F66750E 63745F63 6F64653E 3C666C5F *nct_code=INQ</fl_funcnct_code<fl_*
0100 7265636F 72643E3C 686C5F72 6865793E 30303030 31303030 3C2F666C 5F726865 *records<fl_rkey=0001000</fl_rke*
0120 793E3C66 6C5F636D 706E786D 653E434F 4050414E 5920473C 2F666C5F 636D706E *y><fl_cmpname=COMPANY G</fl_cmpn*
0140 6160653E 3C666C5F 61640772 3E313931 39204059 20535452 4545543C 2F666C5F *ame<fl_addr>1919 MY STREET</fl_*
0160 61640472 3E3C666C 5F706868 6E653E36 31333032 30383833 383C2F66 6C5F7068 *addr<fl_phone>6130208838</fl_ph*
0180 6F6E653E 3C666C5F 62616C6A 6E63653E 30303030 31303030 3C2F666C 5F62616C *one<fl_balance>00001000</fl_bal*
01A0 616E6365 3E3C666C 5F726174 696E673E 413C2F66 6C5F7261 74696E67 3E3C666C *ance<fl_rating>A</fl_rating<fl_*
01C0 5F636F6E * _contact=MR BIGLOIS<fl_contact>*
01E0 3C2F666C *</fl_records<vsam_record_area>*
0200 2F544840 * /TMFW00OperationResponse</SOAP*
0220 2D454E56 *-ENV:Body></SOAP-ENV:Envelope> *
  
```

(1) Data areas are:

- 1 - Parameter list
- 2 - Address/Length pairs of buffers
- 3 - First buffer (max. 4000)
- 4 - second buffer (max. 4000)

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



Here we show task #138, which is the CPIH transaction executing our provider application program, THMFW00. This is the send of the RESPONSE message back to the requester application. Data area 1- is the parameter list, not well documented. Data area 2- is a list of buffer address/length pairs. Data area 3- is the data at the first buffer address in the list, maximum of 4000 bytes shown. Data area 4- is the data at the second buffer address in the list. We have covered these areas in more detail on an earlier slide. The key difference here is that it contains the response message format you specified in the DFHLS2WS job in the SOAP body. Also, a slightly different HTTP header is being returned, which is documented in an HTTP manual.

Summary

- We first took a look at the trace options and resources.
- Next we covered the format of trace entries
- We then covered the use of exception trace entries
- Last we reviewed a few different application profiling opportunities
- Throughout we high-lighted some problem and performance opportunities

- Thank you for your attendance and interest in this topic.
- Should you have any questions or if I can be of assistance in the future, please do not hesitate to contact me.
 - Thomas (Tom) Dunlap
 - tomd@themisinc.com
 - CTO at Themis, Inc.
 - Twitter: @ThemisTom