

Profiling a CICS Transaction Using Trace

Ezriel Gross – Rocket Software

Tuesday, February 24, 10:30am - 11:30am

Location: Salon 17

Session: TECH_324

Agenda

Trace domain

- Trace Table
- SIT parameters
- Type of trace
- Trace points and destination
- Abbreviated, full and short
- Trace components
- CETR overview
- Printing trace
- Profiling a transaction
- Determining EXEC CICS command offsets using trace

Trace table

- The trace table contains footprints of system activity
- A collection of entries represents the start and end of events
- Events are in time sequence
- Trace table entries show the entire execution of a task
- Tracing a task eliminates guesswork about the task's flow of execution and what went wrong

Trace SIT parameters

AUXTR auxiliary trace option

Controls whether any of the three types of CICS trace entry are written to the auxiliary trace data set

INTTR internal trace option

Controls whether any of the three types of CICS trace entry are written to the internal trace. Shown in both system and transaction dumps

GTFTTR generalized trace facility (GTF)

To use the GTF data sets for CICS trace data, the GTF task must be started and running. Same GTF destinations can be shared between programs (VTAM, CICS)

SYSTR CICS system trace records

Note: Setting the master trace lag OFF affects only standard tracing and has no effect on special tracing, which is controlled separately by SPCTR or SPCTRxx trace levels and the CETR transaction

SIT parameters

AUXTR auxiliary trace option

AUXTR=OFF|ON

INTTR internal trace option

INTTR=ON|OFF

GTFTTR generalized trace facility (GTF)

GTFTTR=OFF|ON

SYSTR CICS system trace records

SYSTR=ON|OFF

Trace SIT parameters

USERTR USER trace records

If the user trace lag is off, the user trace facility is disabled, and EXEC CICS ENTER TRACENUM commands receive an INVREQ condition if EXCEPTION is not specified

AUXTRSW auxiliary trace autoswitch facility

ALL permits continuous switching between the DFHAUXT and DFHBUXT datasets

TRTABSZ specifies the size, in kilobytes, of the internal trace table

Minimum of 12 MB, extendable in 4 KB increments
CICS can obtain 64-bit (above-the-bar) storage, rather than 31-bit (above-the-line) storage for the internal trace table, depending on the version of the z/OS operating system, and whether the CICS region operates with transaction isolation

SIT parameters

USERTR USER trace records

USERTR=ON|OFF

AUXTRSW auxiliary trace autoswitch facility

AUXTRSW=NO|ALL|NEXT

TRTABSZ size of the internal trace table

TRTABSZ=12MB| up to 1Gbyte

Trace SIT parameters

TRTRANSZ specifies the size of the transaction dump trace table

TRTRANSZ=1024 | number-of-kilobytes

When a transaction dump is taken, CICS obtains MVS storage in 64-bit storage for the transaction dump trace table

Exception trace records

Are always made and are not controlled by a system initialization parameter

SIT parameters

TRTRANSZ transaction dump trace table

TRTRANSZ=1024 | number-of-kilobytes

Types of trace

- User-controlled
 - Normal tracing
 - Trace points and levels
 - Standard, special, and suppressed tracing
 - Specified by terminal and transaction
 - VTAM exit tracing
 - Requires GTF trace to be active
 - User-exception tracing
- Automatic
 - Exception tracing
 - Program check and abend tracing
 - XRF tracing

User-controlled tracing

- Normal tracing
 - is done by the trace domain
 - each component has various levels of tracing which can be active, and which you can set or suppress individually
 - also, trace can be activated for individual transactions or terminals
- GTF tracing
 - can be used for CICS trace data
 - must be used with the VTAM exit trace
- User-exception tracing
 - To make a user program create exception trace entries, add the EXCEPTION option to an EXEC CICS ENTER TRACENUM command
 - These entries (identified by character string *EXCU) are written to the internal trace table even if internal tracing is turned off
 - Exception tracing cannot be disabled, even if normal tracing is off

Trace points and levels

- Trace points are predefined points in CICS code
- Trace point levels
 - Level 1
 - Level 2
 - Level 3 and 4
- Mainline trace entry points – have a level attribute
- Exception trace entry points
 - have no level specification
 - cannot be disabled or suppressed

Trace points and levels

- The level attribute specifies the level of details to be provided by the trace program. Higher number = more info
 - 1. enough diagnostic info to resolve user errors
 - 2. provide info to resolve errors in CICS code and application code
 - 3 and 4. For special cases, usually to provide information to the support centre. 3 and 4 can be used for very few components.
- Trace point levels can vary. Most mainline trace points have a level of 1 or 2. Only a few have level 3 or 4 or higher.

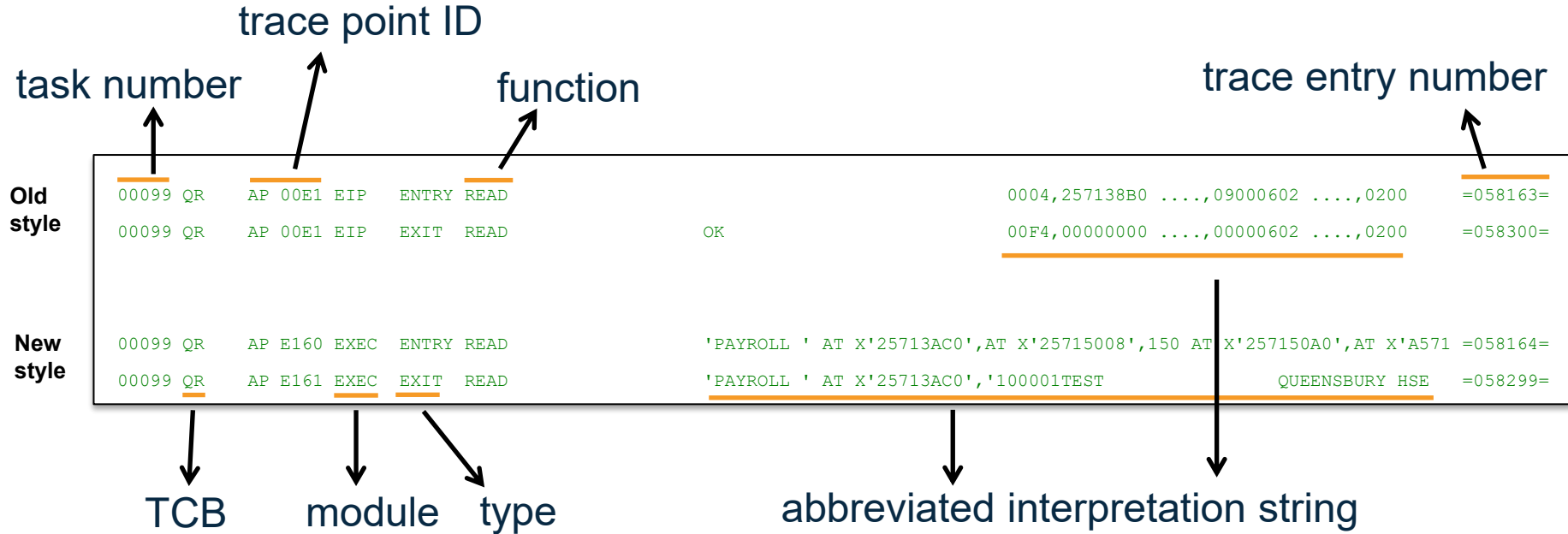
Trace destination

- Internal
 - Wrap-around table in main storage (above the bar)
 - Shown in both system and transaction dumps
 - Minimum of 16 KB, extendable in 4KB increments, max 1 gigabyte
 - Always allocated
- Auxiliary
 - One or two CICS-owned BSAM data sets
 - May be auto-switched
 - Can be set so entries do not wrap
- GTF
 - Written to MVS-defined destination
 - Same GTF destination may be shared between programs (VTAM, CICS)
 - Entries may wrap

Format of trace entries

- Abbreviated (ABBREV)
 - Prints one line per entry
- Short
 - Prints one or multiple lines
- Extended
 - Prints up to 4 KB per entry
 - The terms 'extended' and 'full' mean the same thing in this context

Abbreviated trace entries example



Abbreviated trace entries example

- Task number uniquely identifies a task for as long as it is in the system. It can be:
 - a 5 digit decimal number
 - a 3 character non-numeric value
 - a 2 character domain index
- TCB has the name of the TCB that the trace entry was made from
- Trace point indicates where the trace point is in CICS code. AP 00E1 is an exec interface request in the Application domain
- The [Trace Entries](#) topic in IBM Knowledge Center documents the trace point IDs

Abbreviated trace entries example

- Abbreviated interpretation string shows:
 - the CICS module from which the trace point was made
 - the function being performed
 - any parameters passed on a call, and any responses from a called routine
- Trace entry number is the number of the entry on the trace data set. You can match the abbreviated and the extended trace entries with this number
- The types ENTRY and EXIT are paired. The EXIT reveals if the command was successful
- Event, *EXC* (exception), and *EXCU* (user exception) traces are non-paired types

Full trace – old style and new style

Old style

```
AP 00E1 EIP ENTRY READ                                REQ(0004) FIELD-A(257138B0 ....) FIELD-B(09000602 ....) BOUNDARY(0200)
TASK-00099 KE_NUM-008E TCB-QR /008C6E88 RET-A5597D22 TIME-02:49:40.3761297734 INTERVAL-00.0000061040 =058163=

AP 00E1 EIP EXIT READ OK                             REQ(00F4) FIELD-A(00000000 ....) FIELD-B(00000602 ....) BOUNDARY(0200)
TASK-00099 KE_NUM-008E TCB-QR /008C6E88 RET-A5597D22 TIME-02:49:40.4304042294 INTERVAL-00.0000002714 =058300=

AP E160 EXEC ENTRY READ FILE('PAYROLL ' AT X'25713AC0') INTO( AT X'25715008') LENGTH(150 AT X'257150A0') RIDFLD( AT X'A5714D54')
EQUAL NOHANDLE COBOLII STMT_#(00097)
```

New style

```
TASK-00099 KE_NUM-008E TCB-QR /008C6E88 RET-800850A2 TIME-02:49:40.3761330527 INTERVAL-00.0000032792 =058164=
1-0000 00560000 00162571 3AA80000 00000602 F0002700 008000F0 F0F0F9F7 00010110 *.....y.....0.....00097....*
0020 25713AC0 1B182571 D7C1E8D9 D6D3D340 0002A008 25715008 95A34040 0103030A *...{....PAYROLL ...&.nt ....*
0040 257150A0 4F904040 00960004 0408A571 4D544040 4040 *..&|. .o...v.( *
2-0000 25713AA8 25713AC0 25715008 257150A0 A5714D54 *...y...{...&.v.( *
3-0000 25713A90 *... *
4-0000 0024940F 0116179F C4C1E3F1 0000099C C3D7F6F2 0000017E 00007D34 14000000 *..m....DAT1....CP62...='.....*
0020 00000000 00000000 00000000 00000000 00000040 40404040 00000000 *..... *
0040 00000000 00000000 00000000 00000000 00000000 00 *..... *
5-0000 257138B0 *... *
6-0000 00104001 25713708 40404040 A5597D22 00000000 25714BDC 25713A90 25714D54 *.. .. v.'.....(.*
0020 25597AD0 25597060 40000000 238205EC 00000000 25597160 25714AA0 25714D50 *..:)...- ...b.....-.&.(*
0040 25597718 257129D8 *.....Q

AP E161 EXEC EXIT READ FILE('PAYROLL ' AT X'25713AC0') INTO('100001TEST QUEENSBURY HSE BRIGHTON
SUSSEX 7552990061234.' AT X'25715008') LENGTH(150 AT X'257150A0') RIDFLD( AT X'A5714D54') EQUAL RESP
(O) RESP2(0) NOHANDLE COBOLII STMT_#(00097)

TASK-00099 KE_NUM-008E TCB-QR /008C6E88 RET-80087398 TIME-02:49:40.4304039580 INTERVAL-00.0000056186 =058299=
1-0000 00B80000 00162571 3AA80000 00000602 F0002700 008000F0 F0F0F9F7 00010110 *.....y.....0.....00097....*
0020 25713AC0 1B182571 D7C1E8D9 D6D3D340 0002A06C 25715008 95A34040 F1F0F0F0 *...{....PAYROLL ...&.nt 1000*
0040 F0F1E3C5 E2E34040 40404040 40404040 40404040 4040D8E4 C5C5D5E2 C2E4D9E8 *01TEST QUEENSBURY*
0060 40C8E2C5 40404040 4040C2D9 C9C7C8E3 D6D54040 40404040 40404040 4040E2E4 * HSE BRIGHTON SU*
0080 E2E2C5E7 40404040 40404040 40404040 4040F7F5 F5F2F9F9 F0F04AF1 F2F3F44E *SSEX 7552990061234.*
00A0 0103030A 257150A0 0002C100 00960004 0408A571 4D540B88 0000 *.....&...A...o...v.(...h..*
2-0000 25713AA8 25713AC0 25715008 257150A0 A5714D54 *...y...{...&.v.( *
3-0000 25713A90 *... *
4-0000 0024940F 0116179F C4C1E3F1 0000099C C3D7F6F2 0000017E 00007D06 02000000 *..m....DAT1....CP62...='.....*
0020 000000D7 C1E8D9D6 D3D34000 00000000 000000D7 C1E8D9D6 D3D34000 00000000 *...PAYROLL .....PAYROLL .....*
0040 00000000 00000000 00000000 00000000 00000000 00 *..... *
```

Full trace – old style and new style

Old style

```
AP 00E1 EIP EXIT READ OK                                REQ(00F4) FIELD-A(00000000 ....) FIELD-B(00000602 ....) BOUNDARY(0200)
                TASK-00099 KE_NUM-008E TCB-QR /008C6E88 RET-A5597D22 TIME-02:49:40.4304042294 INTERVAL-00.0000002714 =058300=
```

trace point ID function



New style

```
AP E161 EXEC EXIT READ FILE('PAYROLL ' AT X'25713AC0') INTO('100001TEST                QUEENSBURY HSE                BRIGHTON
                SUSSEX                75529900&1234.' AT X'25715008') LENGTH(150 AT X'257150A0') RIDFLD( AT X'A5714D54') EQUAL RESP
                (0) RESP2(0) NOHANDLE COBOLII STMT_#(00097)

TASK-00099 KE_NUM-008E TCB-QR /008C6E88 RET-80087398 TIME-02:49:40.4304039580 INTERVAL-00.0000056186 =058299=
1-0000 00BĀ0000 00162571 3AA80000 00000602 F0002700 008000F0 F0F0F9F7 00010110 *.....y.....0.....00097....*
0020 25713AC0 1B182571 D7C1E8D9 D6D3D340 0002A06C 25715008 95A34040 F1F0F0F0 *...{...PAYROLL ...%.&.nt 1000*
0040 F0F1E3C5 E2E34040 40404040 40404040 40404040 4040D8E4 C5C5D5E2 C2E4D9E8 *01TEST                QUEENSBURY*
0060 40C8E2C5 40404040 4040C2D9 C9C7C8E3 D6D54040 40404040 40404040 4040E2E4 * HSE                BRIGHTON                SU*
0080 E2E2C5E7 40404040 40404040 40404040 4040F7F5 F5F2F9F9 F0F04AF1 F2F3F44B *SSEX                75529900&1234.*
00A0 0103030A 257150A0 0002C100 00960004 0408A571 4D540B88 0000                *.....&...A..o.....v.(.h.. *
2-0000 25713AA8 25713AC0 25715008 257150A0 A5714D54 *...y...{..&...&.v.( *
3-0000 25713A90 *..... *
4-0000 0024940F 0116179F C4C1E3F1 0000099C C3D7F6F2 0000017E 00007D06 02000000 *..m.....DAT1....CP62...='.....*
0020 000000D7 C1E8D9D6 D3D34000 00000000 000000D7 C1E8D9D6 D3D34000 00000000 *...PAYROLL .....PAYROLL .....*
0040 00000000 00000000 00000000 00000000 00000000 00                *.....*
```

data



Full trace entries example

The full trace contains more comprehensive interpretation

The kernel task number (KE_NUM) is the number used by the KE domain to identify the task
The TCB field contains the TCB name as well as the address of the MVS TCB in use for this task. The Call return address (RET) is the return address passed in R14 to a called routine

For an EXEC CICS call the RET field becomes invaluable

By subtracting the storage address of where the program is loaded from the return value, you can determine the offset in the program of the executed instruction

It is also possible to determine if a handle condition was invoked by looking at this value. If this address is not the same on an entry trace record as the corresponding exit trace record, a handle condition was invoked

Timestamp

Interval is the interval since the last trace entry

Exception trace

abbrev.

```
00099 QR    AP 0781 SRP    *EXC* ABEND_ASRA                DATABUS1,00000DA8,USER                =058309=
```

short

```
00099 QR    AP 0781 SRP    *EXC* ABEND_ASRA PROGRAM(DATABUS1) OFFSET(00000DA8) EXEC_KEY(USER)
                                           RET-A268E614 02:49:40.4374540776 00.0000005341 =058309=
```

full

```
AP 0781 SRP *EXC* - ABEND_ASRA PROGRAM(DATABUS1) OFFSET(00000DA8) EXEC_KEY(USER)

TASK-00099 KE NUM-008E TCB-QR    /008C6E88 RET-A268E614 TIME-02:49:40.4374540776 INTERVAL-00.0000005341          =058309=
1-0000 C4CIE3C1 C2E4E2F1 F0C3F761 C1D2C5C1 00000DA8 00020781                *DATABUS10C7/AKEA...y...a          *
2-0000 00                                *..                                *
3-0000 F0C3F761 C1D2C5C1 018600C7 00000000 C4C6C8C1 D7D3C9F1 00000000 22E22C00 *0C7/AKEA.f.G...DFHAPLI1.....S...*
0020 00000000 226A0800 00000000 2398E000 42678B00 00000001 00000000 FFFFFFFF *.....q\.....*
0040 07950000 80000000 00000000 25597DA8 00060007 00000000 00000000 00000000 *..n.....'y.....*
0060 90800000 00000000 00000000 25714BDC 00000000 25713A90 00000000 25714CC8 *.....<H*
0080 00000000 00000000 00000000 25700100 00000000 40000000 00000000 238205EC *.....b..*
00A0 00000000 00000000 00000000 25597160 00000000 25714AA0 00000000 25714D50 *.....-.....ç.....(&*
00C0 00000000 25597718 00000000 25597124 00000000 257138B0 00000000 A5597D22 *.....v.'.*
00E0 00000000 00000000 0007FFFF 00000000 00000000 00000000 00000000 00000000 *.....*
0100 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0120 00000000 00000000 07950000 80000000 00000000 25597DA8 00060007 00000000 *.....n.....'y.....*
0140 00000000 00000000 90800000 00000000 00000000 25714BDC 00000000 25713A90 *.....*
0160 00000000 25714CC8 00000000 00000000 00000000 25700100 00000000 40000000 *.....<H.....*
0180 00000000 238205EC 00000000 00000000 00000000 25597160 00000000 25714AA0 *.....b.....-.....ç.....*
01A0 00000000 25714D50 00000000 25597718 00000000 25597124 00000000 257138B0 *.....(&.....*
01C0 00000000 A5597D22 00000000 00000000 0007FFFF 00000000 00000000 00000000 *.....v.'.....*
01E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0200 00000000 00000000 00000000 00000000 D0F43268 0C430274 00000000 00000000 *.....}4.....*
0220 41120DD7 50429B6D 00000000 00000000 4116A09E 667F3BCD 00000000 00000000 *...P&..._.....".....*
0240 40B504F3 33F9DE65 00000000 00000000 00000000 00000000 00000000 00000000 *...3.9.....*
0260 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
```

Exception trace entry example

- There is much more detail in the amount of data for full entries compared to abbreviated
- An exception trace entry is a type of extended trace entry. These have EXC in the interpretation string area.
- Exception trace entries are written even if CICS tracing is turned off.
- They occur when an abnormal condition, such as an ASRA abend, occurs

Short trace

A short trace is a full trace without data areas.

Old style

```
00099 QR    AP 00E1 EIP    ENTRY READ                      REQ(0004) FIELD-A(257138B0 ....) FIELD-B(09000602 ....)
                                BOUNDARY(0200)                      RET-A5597D22 02:49:40.3761297734 00.0000061040 =058163=

00099 QR    AP 00E1 EIP    EXIT  READ OK                      REQ(00F4) FIELD-A(00000000 ....) FIELD-B(00000602 ....)
                                BOUNDARY(0200)                      RET-A5597D22 02:49:40.4304042294 00.0000002714 =058300=
```

New style

```
00099 QR    AP E160 EXEC    ENTRY READ FILE('PAYROLL ' AT X'25713AC0') INTO( AT X'25715008') LENGTH(150 AT X'257150A0') RIDFLD( AT
                                X'A5714D54') EQUAL NOHANDLE COBOLII STMT_#(00097)
                                RET-800850A2 02:49:40.3761330527 00.0000032792 =058164=

00099 QR    AP E161 EXEC    EXIT  READ FILE('PAYROLL ' AT X'25713AC0') INTO('100001TEST                      QUEENSBURY HSE          BRIGHTON
                                SUSSEX                          75529900¢1234.' AT X'25715008') LENGTH(150 AT X'257150A0')
                                RIDFLD( AT X'A5714D54') EQUAL RESP(0) RESP2(0) NOHANDLE COBOLII STMT_#(00097)
                                RET-80087398 02:49:40.4304039580 00.0000056186 =058299=
```

Trace selection

- Types of tracing
 - Abbreviated, short, and full trace entries
- Component trace
 - Specify the level of tracing required
- Selection criteria may include
 - Termid
 - Tranid
- Additional tracing options
 - VTAM exit tracing
 - User exception tracing

Trace selection

- **Standard trace**
Standard tracing is done by the trace domain to record execution flow
- **Component trace**
Each component has various levels of tracing which can be set or suppressed individually
- **Specified by terminal and transaction**
By default, trace is turned on for a transaction running at any terminal.
Trace can be activated for individual transactions or terminals.
- **Special trace**
Allows recording of alternative trace entries
- **VTAM Exit tracing**
GTF trace can be used for CICS trace data
- **Exception tracing**
Cannot be suppressed
CICS records Program check and ABEND information automatically

Trace levels

It is possible to select up to 32 levels of tracing. However, most CICS components only use levels 1, 2 and 3, and some do not have trace points at all these levels.

The exceptions are the SM component (storage manager domain), which also has level 4 tracing; and the SJ component (JVM domain), which also has trace levels 29–32.

SIT parameters

STNTR specifies the level of standard tracing

STNTR=1|(1[,2][,3])|ALL|OFF

STNTRxx standard CICS component tracing

STNTRxx=1|(1[,2][,3][,4][,29][,30][,31][,32])|ALL|OFF

SPCTR specifies the level of special tracing

SPCTR=(1,2)|(1[,2][,3])|ALL|OFF

SPCTRxx special component level tracing

SPCTRxx=(1,2)|(1[,2][,3][,4][,29][,30][,31][,32])|ALL|OFF

Trace levels 1-4 and 29-32

Level 1

Provides documented diagnostic information to resolve user errors
Located on entry to, and exit from, every CICS domain or major internal domain function

Level 2

Provides more detailed information

Level 3 and 4

Only use under advice from the Support Center.

Selecting tracing levels 3, 4, or ALL for standard tracing for the storage manager (SM) component, or the temporary storage domain (TS), degrades the performance of your CICS region. This is because options 3 and 4 (and ALL) switch on trace flags that are used by SM domain for field engineering purposes.

Level 29,30,31,32

The SJ component (JVM domain), supports trace levels 29–32, which are reserved to indicate the JVM trace levels 0, 1, and 2, plus a user-definable JVM trace level

Trace component names 1

Code	Component name	Code	Component name
AP	Application domain	DP	Debugging profiles domain
AS	Asynchronous Services domain	DS	Dispatcher domain
BA*	Business application manager	DU	Dump domain
BM*	Basic mapping support	EC*	Event capture and emission
BR*	Bridge	EI*	Exec interface
CP*	Common programming interface	EJ	Enterprise Java domain
DC*	Dump compatibility layer	EM	Event manager domain
DD	Directory manager domain	EP	Event processing domain
DH	Document handling domain	FC*	File control
DM	Domain manager domain	GC	Global catalog domain

Trace component names 2

Code	Component name	Code	Component name
IC*	Interval control	ML	Markup language domain
IE	ECI over TCP/IP domain	MN	Monitoring domain
IS*	ISC or IRC	MP	Managed platform domain
KC*	Task control	NQ	Enqueue domain
KE	Kernel	OT	Object transaction domain
LC	Local catalog domain	PA	Parameter domain
LD	Loader domain	PC*	Program control
LG	Log manager domain	PG	Program manager domain
LM	Lock domain	PI	Pipeline domain
ME	Message domain	PT	Partner domain

Trace component names 3

Code	Component name	Code	Component name
RA	Resource manager adapters	SM	Storage manager domain
RI*	Resource manager interface (RMI)	SO	Sockets domain
RL	Resource life-cycle domain	ST	Statistics domain
RM	Recovery manager domain	SZ*	Front End Programming Interface
RS	Region status domain	TC*	Terminal control
RX	RRS-coordinated EXCI domain	TD*	Transient data
RZ	Request streams domain	TI	Timer domain
SC*	Storage control	TR	Trace domain
SH	Scheduler services domain	TS	Temporary storage domain
SJ	JVM domain	UE*	User exit interface

Trace component names 4

Code	Component name
US	User domain
WB	Web domain
WU	System management RESTful API
W2	Web 2.0 domain
XM	Transaction manager domain
XS	Security manager domain

- Note Components marked with * are subcomponents of the AP Domain. The trace entries for these components are produced with a trace point ID of AP nnnn, where nnnn is unique within the AP domain.

CETR overview

- The CICS-supplied CETR transaction can control trace destinations, component tracing, terminal and transaction tracing. It can also activate or deactivate any destination, set master trace flags, or dynamically alter the size of the trace table
- Pressing PF4 from the first CETR screen displays this slide. The list of CICS components, or domains, is shown with the current settings for standard and special (greater detail) trace levels
- Here you can see the options set for standard and special tracing. This allows you to set two different tracing criteria for each component. The Standard and Special columns indicate the trace point levels
- Pressing PF1 on this screen displays the next screen containing the component abbreviations and the associated meaning

CETR: Component Trace Options

There are 4 pages of component trace options

Use PF8 to page down as normal

CETR		Component Trace Options		T53I	CCVT53I
Over-type Component	Standard	Special		PAGE 1 OF 4	
AP	1	1			
BA	1	1			
BM	1	1			
BR	1	1			
CP	1	1			
DC	1	1			
DD	1	1			
DH	1	1			
DM	1	1			
DP	1	1			
DS	1	1			
DU	1	1			
EC	1	1			
EI	1-2	1-2			
EJ	1	1			
EM	1	1			
EP	1	1			

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

CETR: Transaction and Terminal Trace

```
CETR                               Transaction and Terminal Trace                               T53I CCVT53I

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
```

CETR: Transaction and Terminal Trace

- Use PF5 to display the Transaction and Terminal Trace screen
- Using this screen of the CETR transaction, you can select standard, special or, suppressed tracing for any terminal or transaction. You can also enable the special GTF traces for VTAM exits
 - Standard tracing – normal transaction tracing
 - Special tracing – invoke when transaction or terminal tracing is required, or when a higher level of tracing is needed
 - Suppressed tracing – suppress tracing for a particular transaction or terminal
- If standard tracing is turned off, special tracing can still occur
- To only trace a single transaction or terminal:
 - Turn the master system trace flag off
 - Identify the transaction or terminal you wish to trace

Printing CICS traces

- Transaction dump
 - Uses DFHDUnnn utility
 - Contains only transaction that failed (default)
 - Can be abbreviated followed by full entries
- System dump
 - Formatted with IPCS
 - Shows all transactions
 - Can be abbreviated, short, and full entries
- Auxiliary trace
 - Uses DFHTUnnn utility
 - Can be abbreviated, short, and full entries
 - Can be selective

Auxiliary trace print JCL

```
//*****  
//*  
//* THIS JOB WILL PRINT THE AUXILIARY TRACE DATASET *  
//* -- OPTIONAL SYSIN PARAMETERS *  
//*   ABBREV | SHORT | FULL (DEFAULT) *  
//*   TRANID=(PAY1,DAT1,...) *  
//*   TERMID=(TRM1,TRM2,...) *  
//*   TASKID=(00099-00112,00115,...) *  
//*   TIMERG=(HH MM SS-HH MM SS,...) *  
//*   TYPETR=(AP00E1,APE160-E161,...) *  
//*   KE_NUM=(0097,0098,...) *  
//*   ENTRY NUM=(005781-006439,...) *  
//*   INTERVAL=.300 *  
//*   EXCEPTION *  
//*   ... *  
//*****  
//PRTAUXT EXEC PGM=DFHTU750,REGION=30M  
//STEPLIB DD DISP=SHR,DSN=CICS.V750BASE.CICS.SDFHLOAD  
//DFHAUXT DD DISP=SHR,DSN=CCVCICS.CCVT53I.DFHAUXT  
// DD DISP=SHR,DSN=CCVCICS.CCVT53I.DFHBUXT  
//DFHAXPRM DD SYSOUT=*  
//DFHAXPRM DD *  
//          FULL,TASKID=99,TYPETR=(AP00E1,APE160-E161)  
/*
```

Auxiliary trace print JCL

- Auxiliary trace is printed using the trace utility program
- Abbreviated, short, or full (extended) format
- You can select trace entries by kernel task number, TRANID, TASKID, CICS TASKID, trace ID types, and any time period
- Some additional parameters:
 - ENTRY_NUM allows printing specific trace entries
 - EXCEPTION prints only exception trace entries
 - INTERVAL prints trace entries that exceed the interval specified
 - TYPETR=(AP00E1) shows all the EXEC CICS commands issued and their response. It provides a quick summary of program flow logic.

Profiling a transaction

- Profiling is a technique for determining the behaviour and performance parameters for an online transaction
- Profiling is started at transaction initiation to obtain some basic information such as transaction ID and the first program to be started
- Every EXEC CICS command can be bracketed and responses received
- Obtain fast overview of transaction flow
- Minimum entries need to be reviewed
- Focus on EXEC CICS commands
 - repetitive commands issued
 - non-normal responses
 - resources accessed
 - intervals longer than expected

Profile – page 1 of 3

```
00099 QR    AP 0591 APXM  EXIT  INIT_XM_CLIENT/OK                               =057961=  
00099 QR    AP 1790 TFXM  ENTRY INIT_XM_CLIENT           23A86570 , 02A00000          =057962=  
00099 QR    XM 1001 XMIQ  ENTRY SET_TRANSACTION         TERMINAL,23A86570          =057963=  
00099 QR    XM 1002 XMIQ  EXIT  SET_TRANSACTION/OK      =057964=  
00099 QR    AP 1791 TFXM  EXIT  INIT_XM_CLIENT/OK      00000004,00000000,YES,NO,NO =057965=  
00099 QR    US 0401 USXM  ENTRY INIT_TRANSACTION_USER  00000004,YES                =057966=  
00099 QR    DD 0301 DDLO  ENTRY LOCATE                  22616BC0,23990DD4,USD2,00000004 =057967=  
00099 QR    DD 0302 DDLO  EXIT  LOCATE/OK              22D01400 , 01010101        =057968=  
00099 QR    XS 0401 XSXM  ENTRY ADD_TRANSACTION_SECURITY 426005F0 , 00000004        =057969=  
00099 QR    XS 0402 XSXM  EXIT  ADD_TRANSACTION_SECURITY/OK =057970=  
00099 QR    US 0402 USXM  EXIT  INIT_TRANSACTION_USER/OK 22D0141F , 22D03110,0      =057971=  
00099 QR    DS 0002 DSAT  ENTRY SET_PRIORITY           1                            =057972=  
00099 QR    DS 0003 DSAT  EXIT  SET_PRIORITY/OK       1                            =057973=  
00099 QR    DP 0900 DPXM  ENTRY INIT_XM_CLIENT         =057974=  
00099 QR    DP 0901 DPXM  EXIT  INIT_XM_CLIENT/OK      =057975=  
00099 QR    RM FA01 RMUC  ENTRY CREATE_UOW             NO,BACKWARD,0              =057976=  
00099 QR    RM 0209 RMUC  EVENT Remote_UOW_id_created 160DC6E3E2F14BE2C3F0E3C3D7F6F2F43267FB72EF000100000000 =057977=  
00099 QR    RM FA02 RMUC  EXIT  CREATE_UOW/OK         =057978=  
00099 QR    PI 0B00 PIXM  ENTRY INIT_XM_CLIENT         =057979=  
00099 QR    PI 0B01 PIXM  EXIT  INIT_XM_CLIENT/OK      =057980=  
00099 QR    XS 0701 XSRC  ENTRY CHECK_CICS_RESOURCE    DAT1,TRANSATTACH,EXECUTE   =057981=  
00099 QR    XS 0709 XSRC  EVENT CHECK                   DAT1 CHECK_RESOURCE_ACCESS,426005F0 , 00000004,TCICSTRN,READ,YES,2399 =057982=  
00099 QR    XS 070A XSRC  EVENT CHECK-COMPLETE         DAT1 EJG CHECK_RESOURCE_ACCESS,OK,0,0,0,0 =057983=  
00099 QR    XS 0702 XSRC  EXIT  CHECK_CICS_RESOURCE/OK =057984=  
00099 QR    AP 0590 APXM  ENTRY BIND_XM_CLIENT         =057985=  
00099 QR    AP 0591 APXM  EXIT  BIND_XM_CLIENT/OK      =057986=
```

... cut pages to save trees ...

.
.
.

Profile – page 2 of 3

```
.
00099 QR PG 0901 PGPG ENTRY INITIAL_LINK DATAPGM1 =058017=
00099 QR LD 0001 LLDL ENTRY ACQUIRE_PROGRAM 00000050_40CC6990 =058018=
00099 QR LD 0002 LLDL EXIT ACQUIRE_PROGRAM/OK A5511028,25511000,3A78,REUSABLE,ESDSA,OLD_COPY,,CICSWS,NO,NO =058019=
00099 QR PG 1700 PGCH ENTRY SET_CURRENT_CHANNEL 253EA030,YES =058020=
00099 QR PG 1701 PGCH EXIT SET_CURRENT_CHANNEL/OK =058021=
00099 QR AP 1940 APLI ENTRY START_PROGRAM DATAPGM1,CEDF,FULLAPI,EXEC,NO,253B579C,00000000 , 00000000,1,NO =058022=
00099 QR AP 1948 APLI EVENT CALL-TO-LE/370 Thread_Initialization DATAPGM1 =058023=
00099 QR AP 1949 APLI EVENT RETURN-FROM-LE/370 Thread_Initialization OK DATAPGM1 =058024=
00099 QR SM 0301 SMGF ENTRY GETMAIN 8BA0,YES,LE_RUWA,TASK31 =058025=
00099 QR SM 0302 SMGF EXIT GETMAIN/OK 25705528 =058026=
00099 QR AP 1948 APLI EVENT CALL-TO-LE/370 Rununit_Init_&_Begin_Invo DATAPGM1 =058027=
00099 QR AP 00E1 EIP ENTRY ASKTIME-ABSTIME 0004,2570AD00 .....,09004A02 .....,0200 =058028=
00099 QR AP E160 EXEC ENTRY ASKTIME AT X'A570C448',COBOLII,00085 =058029=
00099 QR TI 0200 TIMF ENTRY INQUIRE_TIME LOCAL,YES =058030=
00099 QR TI 0201 TIMF EXIT INQUIRE_TIME/OK 2E9A1E,003675984580368C,02,49,40,368 =058031=
00099 QR AP D500 UEH EVENT LINK-TO-USER-EXIT-PROGRAM GLUEEXIT AT EXIT POINT XEIOUT =058032=
00099 QR AP D501 UEH EVENT RETURN-FROM-USER-EXIT-PROGRAM GLUEEXIT WITH RETURN CODE 0 =058033=
00099 QR AP E161 EXEC EXIT ASKTIME 3675984580368 AT X'A570C448',0,0,COBOLII,00085 =058034=
00099 QR AP 00E1 EIP EXIT ASKTIME-ABSTIME OK 00F4,00000000 .....,00004A02 .....,0200 =058035=
00099 QR AP 00E1 EIP ENTRY FORMATTIME 0004,2570AD00 .....,09004A04 .....,0200 =058036=
00099 QR AP E160 EXEC ENTRY FORMATTIME 3675984580368 AT X'2570C448',AT X'2570C42B',X'61' AT X'2570AF30',AT X =058037=
00099 QR TI 0200 TIMF ENTRY FORMAT_TIME 003675984580368C =058038=
00099 QR TI 0201 TIMF EXIT FORMAT_TIME/OK 7E0,6,1B,170,B3,1,Sun, 26 Jun 2016 18:49:40 +0000,2016-06-26T18:49:40 =058039=
00099 QR AP D500 UEH EVENT LINK-TO-USER-EXIT-PROGRAM GLUEEXIT AT EXIT POINT XEIOUT =058040=
00099 QR AP D501 UEH EVENT RETURN-FROM-USER-EXIT-PROGRAM GLUEEXIT WITH RETURN CODE 0 =058041=
00099 QR AP E161 EXEC EXIT FORMATTIME 3675984580368 AT X'2570C448','27/06/16' AT X'2570C42B',X'61' AT X'257 =058042=
00099 QR AP 00E1 EIP EXIT FORMATTIME OK 00F4,00000000 .....,00004A04 .....,0200 =058043=
.
```

```

.
00099 QR AP 00E1 EIP ENTRY GET-CONTAINER 0004,2570AD00 ....,09003414 ....,0200 =058044=
00099 QR AP E160 EXEC ENTRY GET 'request-cont ' AT X'2570AF40','payroll ' AT X'A570C180',A =058045=
00099 QR AP F801 EIBAM ENTRY GET CONTAINER =058046=
00099 QR PG 1700 PGCH ENTRY INQUIRE_CHANNEL payroll =058047=
00099 QR PG 1701 PGCH EXIT INQUIRE_CHANNEL/OK 253EE030 =058048=
00099 QR PG 1900 PGCR ENTRY GET_CONTAINER INTO 253EE030,request-cont,EXEC,2570C190 , 00000000 , 00000240 =058049=
00099 QR PG 1901 PGCR EXIT GET_CONTAINER INTO/OK ANY,CHAR,2570C190 , 00000240 , 00000240,1,1,25 =058050=
00099 QR AP F802 EIBAM ENTRY GET_CONTAINER RESP=0 RESP2=0 =058051=
00099 QR AP D500 UEH EVENT LINK-TO-USER-EXIT-PROGRAM GLUEEXIT AT EXIT POINT XEIOUT =058052=
00099 QR AP D501 UEH EVENT RETURN-FROM-USER-EXIT-PROGRAM GLUEEXIT WITH RETURN CODE 0 =058053=
00099 QR AP E161 EXEC EXIT GET 'request-cont ' AT X'2570AF40','payroll ' AT X'A570C180',A =058054=
00099 QR AP 00E1 EIP EXIT GET-CONTAINER OK 00F4,00000000 ....,00003414 ....,0200 =058055=
00099 QR AP 00E1 EIP ENTRY RECEIVE-MAP 0004,2570AD00 ....,09001802 ....,0200 =058056=
00099 QR AP E160 EXEC ENTRY RECEIVE 'DATAM1 ' AT X'2570AED0',AT X'2570C468','DATAMAP' AT X'A570AEE0',TERM =058057=
00099 QR AP 00FA BMS ENTRY MAP-FROM IN MAP MAPSET 0003,00020505 ....,00000020 .... =058058=
00099 QR AP 00FA BMS ENTRY MAP-FROM IN MAP MAPSET 0003,00020505 ....,00000020 .... =058059=
00099 QR PG 0601 PGLD ENTRY LOAD DATAMAPM,TASK LIFE,MAPSET,NO =058060=
00099 QR DD 0301 DDLO ENTRY LOCATE 2263EF30,23993554,PPT,DATAMAPM =058061=
00099 QR DD 0302 DDLO EXIT LOCATE/EXCEPTION NOT FOUND,00800000 , 00000000 =058062=
00099 QR PG 0602 PGLD EXIT LOAD/EXCEPTION PROGRAM_NOT_DEFINED,253E4000,00000000,2398E000 =058063=
00099 QR PG 0601 PGLD ENTRY LOAD DATAMAP,TASK LIFE,MAPSET,NO =058064=
00099 QR DD 0301 DDLO ENTRY LOCATE 2263EF30,23993554,PPT,DATAMAP =058065=
00099 QR DD 0302 DDLO EXIT LOCATE/OK D7D7E3C5 , 23DCDED0 =058066=
00099 QR LD 0001 LDLD ENTRY ACQUIRE_PROGRAM YES,00000050 40CC6468 =058067=
00099 QR LD 0002 LDLD EXIT ACQUIRE_PROGRAM/OK 80075000,00075000,1F90,REUSABLE,CDSA,OLD_COPY,,CICSW,NO,NO =058068=
00099 QR PG 0602 PGLD EXIT LOAD/OK 00075000,80075000,1F90 =058069=
00099 QR SM 0C01 SMMG ENTRY GETMAIN 13C,23A86570,YES,00,TERMINAL =058070=
00099 QR SM 0C02 SMMG EXIT GETMAIN/OK 253E4120 =058071=
00099 QR SM 0D01 SMMF ENTRY FREEMAIN 253E4000,23A86570 =058072=
00099 QR SM 0D02 SMMF EXIT FREEMAIN/OK TERMINAL storage at 253E4000 =058073=
00099 QR AP 00FA BMS EXIT 0005,00000000 ....,00000000 .... =058074=
00099 QR AP 00FA BMS EXIT 0005,00000000 ....,00000000 .... =058075=
00099 QR SM 0D01 SMMF ENTRY FREEMAIN 253E4120,23A86570 =058076=
00099 QR SM 0D02 SMMF EXIT FREEMAIN/OK TERMINAL storage at 253E4120 =058077=
00099 QR AP D500 UEH EVENT LINK-TO-USER-EXIT-PROGRAM GLUEEXIT AT EXIT POINT XEIOUT =058078=
00099 QR AP D501 UEH EVENT RETURN-FROM-USER-EXIT-PROGRAM GLUEEXIT WITH RETURN CODE 0 =058079=
00099 QR AP E161 EXEC EXIT RECEIVE 'DATAM1 ' AT X'2570AED0',AT X'2570C468','DATAMAP' AT X'A570AEE0',TERM =058080=
00099 QR AP 00E1 EIP EXIT RECEIVE-MAP OK 00F4,00000000 ....,00001802 ....,0200 =058081=

```

Alternate profile – application events

C:\PROF Application Events

Command ==>

Row 1 of 638 More: >

Scroll ==> CSR

Tran: DAT1 Start: 2016-06-27 02:49:40.368042 Response: 3.459826 Task: 99

/	Relative	Program	Call	EIBRESP	Command
---	+0.000000	DATAPGM1	ATTACH TASK		ATTACH/OK TRANSACTION TOKEN(253DC300 , 0000099C) TRANNUM(0000099C)
---	+0.000310	DATAPGM1	START PROGRAM		START_PROGRAM PROGRAM(DATAPGM1) CEDF_STATUS(CEDF) EXECUTION_SET(FULLAPI) ENVIRONMENT_TYPE(EXEC) SYNCONRETURN(NO)
---	+0.000404	DATAPGM1	ASKTIME ABSTIME	OK	ASKTIME ABSTIME(3675984580368) COBOLII STMT_#(00085)
---	+0.000440	DATAPGM1	FORMATTIME	OK	FORMATTIME ABSTIME(3675984580368) DMMYY('27/06/16') DATESEP(X'61') TIME('02:49:40') TIMESEP(X'7A') COBOLII STMT
---	+0.000462	DATAPGM1	GET CONTAINER	OK	GET CONTAINER('request-cont') CHANNEL('payroll') INTO(AT X'2570C190') FLENGTH(576) COBOLII STMT_#(00103)
---	+0.000491	DATAPGM1	RECEIVE MAP	OK	RECEIVE MAP('DATAMI') INTO(AT X'2570C468') MAPSET('DATAMAP') TERMINAL COBOLII STMT_#(00271)
---	+0.000555	DATAPGM1	PUT CONTAINER	OK	PUT CONTAINER('request-cont') CHANNEL('payroll') FROM(AT X'2570C190') FLENGTH(576) DATATYPE(1019) COBOLII STMT_#
---	+0.000572	DATAPGM1	LINK		LINK PROGRAM('DATABUS1') CHANNEL('payroll') COBOLII STMT_#(00416)
---	+0.008053	DATABUS1	ASSIGN	OK	ASSIGN COBOLII STMT_#(00053)
---	+0.008065	DATABUS1	GET CONTAINER	OK	GET CONTAINER('request-cont') CHANNEL('payroll') INTO(AT X'25714D50') FLENGTH(576) COBOLII STMT_#(00056)
---	+0.008086	DATABUS1	READ	OK	READ FILE('PAYROLL') INTO('100001TEST QUEENSBURY HSE BRIGHTON SUSSEX 7552990041234.') LENGTH(150) RIDFLD(AT X'A5
---	+3.221692	DATABUS1	ABEND	*ASRA*	UPDATE_ABEND RECORD ERROR OFFSET(DA8) ABEND_TOKEN(253E4008) FAILING_PROGRAM(DATABUS1) STOKEN(4022BDB0000002F) A
---	+3.225361	DATABUS1	GETMAIN	OK	GETMAIN SET(X'25716CA8') FLENGTH(14904) SYSEIB ASM STMT_#(00000460)
---	+3.225411	DATABUS1	GETMAIN	OK	GETMAIN SET(X'2571A6F8') FLENGTH(4080) SYSEIB ASM STMT_#(00000460)
---	+3.225432	DATABUS1	GETMAIN	OK	GETMAIN SET(X'2571B6F8') FLENGTH(7232) SYSEIB ASM STMT_#(00000460)
---	+3.225540	DATABUS1	GETMAIN	OK	GETMAIN SET(X'2571D348') FLENGTH(4080) SYSEIB ASM STMT_#(00000460)
---	+3.225649	DATABUS1	ADDRESS	OK	ADDRESS SYSEIB ASM STMT_#(00000247)
---	+3.225657	DATABUS1	LOAD	OK	LOAD PROGRAM('CEEMENU3') SET(X'26157000') FLENGTH(54128) ENTRY(X'A6157000') SYSEIB ASM STMT_#(00000285)
---	+3.250039	DATABUS1	GETMAIN	OK	GETMAIN SET(X'2571E348') FLENGTH(4080) SYSEIB ASM STMT_#(00000460)
---	+3.250066	DATABUS1	WRITEQ TD	OK	WRITEQ TD QUEUE('CESE') FROM(' CP62DAT1 20160627024943 CEE3207S The system detected a data exception(System Comp
---	+3.250149	DATABUS1	WRITEQ TD	OK	WRITEQ TD QUEUE('CESE') FROM(' CP62DAT1 20160627024943 From compile unit DATABUS at entry point DATABUS at compi
---	+3.250167	DATABUS1	WRITEQ TD	OK	WRITEQ TD QUEUE('CESE') FROM(' CP62DAT1 20160627024943 25597DA2.') LENGTH(43) SYSEIB ASM STMT_#(00000412)
---	+3.250201	DATABUS1	GETMAIN	OK	GETMAIN SET(X'2571F348') FLENGTH(4080) SYSEIB ASM STMT_#(00000460)
---	+3.250240	DATABUS1	GETMAIN	OK	GETMAIN SET(X'25720348') FLENGTH(4080) SYSEIB ASM STMT_#(00000460)
---	+3.250324	DATABUS1	WRITEQ TD	OK	WRITEQ TD QUEUE('CESE') FROM('1CP62DAT1 20160627024943 CEE3DMP V2 R2.0: Condition processing resulted in the unh
---	+3.250343	DATABUS1	WRITEQ TD	OK	WRITEQ TD QUEUE('CESE') FROM(' CP62DAT1 20160627024943 Task Number: 00099 Transaction ID: DAT1') LENGTH(66) SYSE
---	+3.250360	DATABUS1	WRITEQ TD	OK	WRITEQ TD QUEUE('CESE') FROM(' CP62DAT1 20160627024943') LENGTH(26) SYSEIB ASM STMT_#(00000412)
---	+3.250375	DATABUS1	WRITEQ TD	OK	WRITEQ TD QUEUE('CESE') FROM(' CP62DAT1 20160627024943 CEE38451 CEEDUMP Processing started.') LENGTH(61) SYSEIB
---	+3.250388	DATABUS1	GETMAIN	OK	GETMAIN SET(X'25721348') FLENGTH(6856) SYSEIB ASM STMT_#(00000460)
---	+3.250399	DATABUS1	GETMAIN	OK	GETMAIN SET(X'25722E28') FLENGTH(8040) SYSEIB ASM STMT_#(00000460)
---	+3.250411	DATABUS1	GETMAIN	OK	GETMAIN SET(X'25724DA8') FLENGTH(4184) SYSEIB ASM STMT_#(00000460)
---	+3.250435	DATABUS1	WRITEQ TD	OK	WRITEQ TD QUEUE('CESE') FROM(' CP62DAT1 20160627024943') LENGTH(26) SYSEIB ASM STMT_#(00000412)

Alternate profile – trace events

```
C:\PROF Trace Events                                     Record 00000001 More: < >
Command ==>                                             Scroll ==> CSR
                                                         Format ==> SNGL
/ 02:49:40.368534 Level 0 Gap < 000000.008 >
/ Time (LOCAL)
--- 02:49:40.368534 AP 00E1 EIP ENTRY RECEIVE-MAP REQ(0004)
--- 02:49:40.368538 AP E160 EXEC ENTRY RECEIVE MAP('DATAM1') INTO(AT X'2570C468') MAPSET('DATAMAP') TERMINAL COBOLII STMT_#(00271)
--- 02:49:40.368543 AP 00FA BMS ENTRY MAP-FROM IN MAP MAPSET REQ(0003)
--- 02:49:40.368546 AP 00FA BMS ENTRY MAP-FROM IN MAP MAPSET REQ(0003)
--- 02:49:40.368549 PG 0601 PGLD ENTRY LOAD PROGRAM_NAME(DATAMAPM) HOLD_LIFETIME(TASK_LIFE) MODULE_TYPE(MAPSET) SYSTEM_AUTOINSTALL(NO)
--- 02:49:40.368551 DD 0301 DDLO ENTRY LOCATE DIRECTORY_TOKEN(2263EF30) ENTRY_NAME(23993554) DIRECTORY_NAME(PPT) NAME(DATAMAPM)
--- 02:49:40.368553 DD 0302 DDLO EXIT LOCATE/EXCEPTION_REASON(NOT_FOUND) DATA_TOKEN(00800000 , 00000000)
--- 02:49:40.368555 PG 0602 PGLD EXIT LOAD/EXCEPTION_REASON(PROGRAM_NOT_DEFINED) LOAD_POINT(253E4000) ENTRY_POINT(00000000) PROGRAM_LENGTH(2398E000)
--- 02:49:40.368555 PG 0601 PGLD ENTRY LOAD PROGRAM_NAME(DATAMAP) HOLD_LIFETIME(TASK_LIFE) MODULE_TYPE(MAPSET) SYSTEM_AUTOINSTALL(NO)
--- 02:49:40.368557 DD 0301 DDLO ENTRY LOCATE DIRECTORY_TOKEN(2263EF30) ENTRY_NAME(23993554) DIRECTORY_NAME(PPT) NAME(DATAMAP)
--- 02:49:40.368558 DD 0302 DDLO EXIT LOCATE/OK DATA_TOKEN(D7D7E3C5 , 23DCDE0)
--- 02:49:40.368562 LD 0001 LLDL ENTRY ACQUIRE_PROGRAM_SUSPEND(YES) PROGRAM_TOKEN(00000050_40CC6468)
--- 02:49:40.368563 LD 0002 LLDL EXIT ACQUIRE_PROGRAM/OK ENTRY_POINT(80075000) LOAD_POINT(00075000) PROGRAM_LENGTH(1F90) PROGRAM_ATTRIBUTE(REUSABLE) LOCATION
--- 02:49:40.368569 PG 0602 PGLD EXIT LOAD/OK LOAD_POINT(00075000) ENTRY_POINT(80075000) PROGRAM_LENGTH(1F90)
--- 02:49:40.368571 SM 0C01 SMMG ENTRY GETMAIN GET_LENGTH(13C) TCCTE_ADDRESS(23A86570) SUSPEND(YES) INITIAL_IMAGE(00) STORAGE_CLASS(TERMINAL)
--- 02:49:40.368571 SM 0C02 SMMG EXIT GETMAIN/OK ADDRESS(253E4120)
--- 02:49:40.368578 SM 0D01 SMMF ENTRY FREEMAIN ADDRESS(253E4000) TCCTE_ADDRESS(23A86570)
--- 02:49:40.368583 SM 0D02 SMMF EXIT FREEMAIN/OK TERMINAL storage at 253E4000
--- 02:49:40.368583 AP 00FA BMS EXIT REQ(0005)
--- 02:49:40.368583 AP 00FA BMS EXIT REQ(0005)
--- 02:49:40.368586 SM 0D01 SMMF ENTRY FREEMAIN ADDRESS(253E4120) TCCTE_ADDRESS(23A86570)
--- 02:49:40.368587 SM 0D02 SMMF EXIT FREEMAIN/OK TERMINAL storage at 253E4120
--- 02:49:40.368589 AP D500 UEH EVENT LINK-TO-USER-EXIT-PROGRAM GLUEEXIT AT EXIT POINT XEIOUT
--- 02:49:40.368589 AP D501 UEH EVENT RETURN-FROM-USER-EXIT-PROGRAM GLUEEXIT WITH RETURN CODE 0
--- 02:49:40.368594 AP E161 EXEC EXIT RECEIVE MAP('DATAM1') INTO(AT X'2570C468') MAPSET('DATAMAP') TERMINAL COBOLII STMT_#(00271)
--- 02:49:40.368595 AP 00E1 EIP EXIT RECEIVE-MAP OK REQ(00F4)
***** Bottom of Data *****
```

Determining EXEC CICS command offsets using trace

- Locate the desired EIP entry (EXEC CICS command)
- Note the RET address located on the full or short trace
- Find an LDLD trace entry before the EIP command
- Subtract the entry point address in the LDLD from the RET address on the EIP command giving an offset in a program
- Program name is found from the PG entry before the LDLD

- When you subtract the entry point address, if the offset you obtained is larger than the program length, the problem could be:
 - the incorrect LDLD trace entry was used
 - a subtraction error occurred
 - the program was statically-called. A statically-called program does not have an LDLD entry. The trace by itself does not help determine the offset in the program where an instruction was executed
- Also the LDLD trace entry might not be the one immediately preceding the EIP. Maps or subroutines might have been loaded in between the program load and the EIP

Determining EXEC CICS command offsets using trace

- Using the next slide as an example, look at the EXEC CICS FORMATTIME command at trace entry =058043=
- The RET address for this command is A5511B68 (25511B68)
- Looking backwards in the trace, the LDLD trace entry shows the ENTRY POINT(A5511028) - (25511028) – trace entry =058018=
 - subtracting 25511B68 from 25511028 gives an offset of X 'B40'
 - browsing before the LDLD entry looking for the PG trace entry, the name of the program is identified as **DATAPGM1**
- In summary, the command can be found at offset X'B40' in program **DATAPGM1**
- Alternatively, if EI trace level 2 is active the APE161 says “**COBOLII STMT_#(00090)**”. Note: its RET address points to a CICS module

Determining offsets example – page 1 of 3

RET on EIP
is actual
return
address into
the
application
program

```
00099 QR PG 0901 PPGG ENTRY INITIAL_LINK PROGRAM_NAME(DATAPGM1) RET-A225A406 02:49:40.3683298852 00.0000030336 =058017=  
00099 QR LD 0001 LDLD ENTRY ACQUIRE_PROGRAM PROGRAM_TOKEN(00000050_40CC6990) RET-A2921E96 02:49:40.3683356645 00.0000057792 =058018=  
00099 QR LD 0002 LDLD EXIT ACQUIRE_PROGRAM/OK ENTRY_POINT(A5511028) LOAD_POINT(25511000) PROGRAM_LENGTH(3A78) RET-A2921E96 02:49:40.3683416264 00.0000059619 =058019=  
PROGRAM_ATTRIBUTE(REUSABLE) LOCATION(ESDSA) COPY_STATUS(OLD_COPY) FETCH_TIME() LIBRARY REQ(0004) FIELD-A(2570AD00 ...) FIELD-B(09004A02 ..ç.)  
(CICSWS) BUNDLE_INSTALLED_LIB(NO) PRIVATE_LIBRARY(NO) RET-A5511A6E 02:49:40.3684469809 00.00000714936 =058028=  
00099 QR AP 00E1 EIP ENTRY ASKTIME-ABSTIME RET-800850A2 02:49:40.3684546152 00.0000076342 =058029=  
BOUNDARY(0200) REQ(0004) FIELD-A(2570AD00 ...) FIELD-B(09004A02 ..ç.)  
00099 QR AP E160 EXEC ENTRY ASKTIME ABSTIME( AT X'A570C448') COBOLII STMT #(00085) RET-A5511A6E 02:49:40.3684824111 00.0000033413 =058034=  
00099 QR AP E161 EXEC EXIT ASKTIME ABSTIME(3675984580368 AT X'A570C448') RESP(0) RESP2(0) COBOLII STMT #(00085) RET-80087398 02:49:40.3684824111 00.0000033413 =058034=  
00099 QR AP 00E1 EIP EXIT ASKTIME-ABSTIME OK REQ(00F4) FIELD-A(00000000 ...) FIELD-B(00004A04 ..ç.)  
BOUNDARY(0200) RET-A5511A6E 02:49:40.3684826596 00.0000002485 =058035=  
00099 QR AP 00E1 EIP ENTRY FORMATTIME REQ(0004) FIELD-A(2570AD00 ...) FIELD-B(09004A04 ..ç.)  
BOUNDARY(0200) RET-A5511B68 02:49:40.3684832304 00.0000005708 =058036=  
00099 QR AP E160 EXEC ENTRY FORMATTIME ABSTIME(3675984580368 AT X'2570C448') DMMYY( AT X'2570C42B') DATESEP(X'61' AT RET-800850A2 02:49:40.3684896074 00.0000063769 =058037=  
X'2570AF30') TIME( AT X'2570C433') TIMESEP(X'7A' AT X'A570AF60') COBOLII STMT #(00090) RET-800850A2 02:49:40.3684896074 00.0000063769 =058037=  
00099 QR AP E161 EXEC EXIT FORMATTIME ABSTIME(3675984580368 AT X'2570C448') DMMYY('27/06/16' AT X'2570C42B') DATESEP(X'61' AT RET-80087398 02:49:40.3685029902 00.0000042846 =058042=  
X'2570AF30') TIME('02:49:40' AT X'2570C433') TIMESEP(X'7A' AT X'A570AF60') RESP(0) RESP2(0) RET-80087398 02:49:40.3685029902 00.0000042846 =058042=  
COBOLII STMT #(00090) RET-A5511B68 02:49:40.3685033608 00.0000003706 =058043=  
00099 QR AP 00E1 EIP EXIT FORMATTIME OK REQ(00F4) FIELD-A(00000000 ...) FIELD-B(00004A04 ..ç.)  
BOUNDARY(0200) RET-A5511B68 02:49:40.3685033608 00.0000003706 =058043=  
00099 QR AP 00E1 EIP ENTRY GET-CONTAINER REQ(0004) FIELD-A(2570AD00 ...) FIELD-B(09003414 ...) RET-A5511D46 02:49:40.3685058334 00.0000024726 =058044=  
BOUNDARY(0200) RET-A5511D46 02:49:40.3685058334 00.0000024726 =058044=  
00099 QR AP E160 EXEC ENTRY GET_CONTAINER('request-cont ' AT X'2570AF40') CHANNEL('payroll ' AT X'A570C180') INTO( RET-800850A2 02:49:40.3685078320 00.0000019985 =058045=  
AT X'2570C190') FLENGTH(576 AT X'2570AF50') NOHANDLE COBOLII STMT #(00103) RET-800850A2 02:49:40.3685078320 00.0000019985 =058045=  
00099 QR AP E161 EXEC EXIT GET_CONTAINER('request-cont ' AT X'2570AF40') CHANNEL('payroll ' AT X'A570C180') INTO( RET-80087398 02:49:40.3685314414 00.0000037109 =058054=  
AT X'2570C190') FLENGTH(576 AT X'2570AF50') RESP(0) RESP2(0) NOHANDLE COBOLII STMT #(00103) RET-80087398 02:49:40.3685314414 00.0000037109 =058054=  
00099 QR AP 00E1 EIP EXIT GET-CONTAINER OK REQ(00F4) FIELD-A(00000000 ...) FIELD-B(00003414 ...) RET-A5511D46 02:49:40.3685318193 00.0000003779 =058055=  
BOUNDARY(0200) RET-A5511D46 02:49:40.3685318193 00.0000003779 =058055=
```

Determining offsets example – page 2 of 3

```
00099 QR    AP 00E1 EIP    ENTRY RECEIVE-MAP                               REQ(0004) FIELD-A(2570AD00 ....) FIELD-B(09001802 ....)
                                BOUNDARY(0200)                               RET-A5512706 02:49:40.3685348906 00.0000030712 =058056=
00099 QR    AP E160 EXEC  ENTRY RECEIVE MAP('DATAM1 ' AT X'2570AED0') INTO( AT X'2570C468') MAPSET('DATAMAP' AT X'A570AEE0')
                                TERMINAL NOHANDLE COBOLII STMT #(00271) RET-800850A2 02:49:40.3685384096 00.0000035190 =058057=
00099 QR    LD 0001 LDLD  ENTRY ACQUIRE_PROGRAM SUSPEND(YES) PROGRAM_TOKEN(00000050 40CC6468)
                                RET-A2917204 02:49:40.3685626982 00.0000046245 =058067=
00099 QR    LD 0002 LDLD  EXIT ACQUIRE_PROGRAM/OK ENTRY POINT(80075000) LOAD_POINT(00075000) PROGRAM_LENGTH(1F90)
                                PROGRAM_ATTRIBUTE(REUSABLE) LOCATION(CDSA) COPY STATUS(OLD_COPY) FETCH_TIME() LIBRARY
                                (CICSW$) BUNDLE_INSTALLED_LIB(NO) PRIVATE_LIBRARY(NO)
                                RET-A2917204 02:49:40.3685637968 00.0000010986 =058068=
00099 QR    AP E161 EXEC  EXIT RECEIVE MAP('DATAM1 ' AT X'2570AED0') INTO( AT X'2570C468') MAPSET('DATAMAP' AT X'A570AEE0')
                                TERMINAL RESP(0) RESP2(0) NOHANDLE COBOLII STMT #(00271)
                                RET-80087398 02:49:40.3685947929 00.0000049707 =058080=
00099 QR    AP 00E1 EIP    EXIT RECEIVE-MAP OK                               REQ(00F4) FIELD-A(00000000 ....) FIELD-B(00001802 ....)
                                BOUNDARY(0200)                               RET-A5512706 02:49:40.3685950190 00.0000002260 =058081=
00099 QR    AP 00E1 EIP    ENTRY PUT-CONTAINER                               REQ(0004) FIELD-A(2570AD00 ....) FIELD-B(09003416 ....)
                                BOUNDARY(0200)                               RET-A5512CD0 02:49:40.3685982343 00.0000032153 =058082=
00099 QR    AP E160 EXEC  ENTRY PUT CONTAINER('request-cont ' AT X'2570AF50') CHANNEL('payroll ' AT X'2570C180') FROM(
                                AT X'2570C190') FLENGTH(576 AT X'2570AF60') DATATYPE(1019 AT X'A570B048') COBOLII STMT #
                                (00410) RET-800850A2 02:49:40.3686017451 00.0000035107 =058083=
00099 QR    AP E161 EXEC  EXIT PUT CONTAINER('request-cont ' AT X'2570AF50') CHANNEL('payroll ' AT X'2570C180') FROM(
                                AT X'2570C190') FLENGTH(576 AT X'2570AF60') DATATYPE(1019 AT X'A570B048') RESP(0) RESP2(0)
                                COBOLII STMT #(00410) RET-80087398 02:49:40.3686139736 00.0000037348 =058092=
00099 QR    AP 00E1 EIP    EXIT PUT-CONTAINER OK                               REQ(00F4) FIELD-A(00000000 ....) FIELD-B(00003416 ....)
                                BOUNDARY(0200)                               RET-A5512CD0 02:49:40.3686145068 00.0000005332 =058093=
00099 QR    AP 00E1 EIP    ENTRY LINK                                       REQ(0004) FIELD-A(2570AD00 ....) FIELD-B(09000E02 ....)
                                BOUNDARY(0200)                               RET-A5512D86 02:49:40.3686150229 00.0000005161 =058094=
00099 QR    AP E160 EXEC  ENTRY LINK PROGRAM('DATABUS1' AT X'2570AEE8') CHANNEL('payroll ' AT X'A570C180') NOHANDLE COBOLII
                                STMT #(00416) RET-800850A2 02:49:40.3686179477 00.0000029248 =058095=
00099 QR    LD 0001 LDLD  ENTRY ACQUIRE_PROGRAM PROGRAM_TOKEN(00000050_40CC6780)
                                RET-A290D9BC 02:49:40.3686338466 00.0000069160 =058099=
00099 QR    LD 0002 LDLD  EXIT ACQUIRE_PROGRAM/OK ENTRY POINT(A5597028) LOAD_POINT(25597000) PROGRAM_LENGTH(35D8)
                                PROGRAM_ATTRIBUTE(REUSABLE) LOCATION(ESDSA) COPY STATUS(NEW_COPY) FETCH_TIME() LIBRARY
                                (CICSW$) BUNDLE_INSTALLED_LIB(NO) PRIVATE_LIBRARY(NO)
                                RET-A290D9BC 02:49:40.3760121572 00.0000199750 =058126=
```

Determining offsets example – page 3 of 3

```
00099 QR    AP 00E1 EIP    ENTRY ASSIGN                                REQ(0004) FIELD-A(257138B0 ....) FIELD-B(09000208 ....)
                                BOUNDARY(0200)                                RET-A55978CC 02:49:40.3760966655 00.0000270488 =058143=
00099 QR    AP E160 EXEC  ENTRY ASSIGN COBOLII STMT #(00053)          RET-800850A2 02:49:40.3760978261 00.0000011606 =058144=
00099 QR    AP E161 EXEC  EXIT  ASSIGN RESP(0) RESP2(0) COBOLII STMT_#(00053)
                                RET-80087398 02:49:40.3761076816 00.0000023188 =058149=
00099 QR    AP 00E1 EIP    EXIT  ASSIGN OK                                REQ(00F4) FIELD-A(00000000 ....) FIELD-B(00000208 ....)
                                BOUNDARY(0200)                                RET-A55978CC 02:49:40.3761079116 00.0000002299 =058150=
00099 QR    AP 00E1 EIP    ENTRY GET-CONTAINER                                REQ(0004) FIELD-A(257138B0 ....) FIELD-B(09003414 ....)
                                BOUNDARY(0200)                                RET-A5597AA2 02:49:40.3761085087 00.0000005971 =058151=
00099 QR    AP E160 EXEC  ENTRY GET CONTAINER('request-cont ' AT X'25713B30') CHANNEL('payroll ' AT X'A5714F90') INTO(
                                AT X'25714D50') FLENGTH(576 AT X'25713B40') COBOLII STMT_#(00056)
                                RET-800850A2_02:49:40.3761121704 00.0000036616 =058152=
00099 QR    AP E161 EXEC  EXIT  GET CONTAINER('request-cont ' AT X'25713B30') CHANNEL('payroll ' AT X'A5714F90') INTO(
                                AT X'25714D50') FLENGTH(576 AT X'25713B40') RESP(0) RESP2(0) COBOLII STMT_#(00056)
                                RET-80087398 02:49:40.3761217900 00.0000039599 =058161=
00099 QR    AP 00E1 EIP    EXIT  GET-CONTAINER OK                                REQ(00F4) FIELD-A(00000000 ....) FIELD-B(00003414 ....)
                                BOUNDARY(0200)                                RET-A5597AA2 02:49:40.3761236694 00.0000018793 =058162=
00099 QR    AP 00E1 EIP    ENTRY READ                                REQ(0004) FIELD-A(257138B0 ....) FIELD-B(09000602 ....)
                                BOUNDARY(0200)                                RET-A5597D22 02:49:40.3761297734 00.0000061040 =058163=
00099 QR    AP E160 EXEC  ENTRY READ FILE('PAYROLL ' AT X'25713AC0') INTO( AT X'25715008') LENGTH(150 AT X'257150A0') RIDFLD( AT
                                X'A5714D54') EQUAL NOHANDLE COBOLII STMT_#(00097)
                                RET-800850A2 02:49:40.3761330527 00.0000032792 =058164=
00099 QR    AP E161 EXEC  EXIT  READ FILE('PAYROLL ' AT X'25713AC0') INTO('100001TEST QUEENSBURY HSE BRIGHTON
                                SUSSEX 75529900+1234.' AT X'25715008') LENGTH(150 AT X'257150A0')
                                RIDFLD( AT X'A5714D54') EQUAL RESP(0) RESP2(0) NOHANDLE COBOLII STMT_#(00097)
                                RET-80087398 02:49:40.4304039580 00.0000056186 =058299=
00099 QR    AP 00E1 EIP    EXIT  READ OK                                REQ(00F4) FIELD-A(00000000 ....) FIELD-B(00000602 ....)
                                BOUNDARY(0200)                                RET-A5597D22 02:49:40.4304042294 00.0000002714 =058300=
```

Alternate example

C:\PROF Application Events

Row 1 of 638 More: >

Command ==>

Scroll ==> [CSR](#)

Tran: DAT1 Start: 2016-06-27 02:49:40.368042 Response: 3.459826 Task: 99

```
/ Program EIBRESP Command
---
DATAPGM1 ATTACH/OK TRANSACTION_TOKEN(253DC300 , 0000099C) TRANNUM(0000099C)
DATAPGM1 START_PROGRAM PROGRAM(DATAPGM1) CEDF STATUS(CEDF) EXECUTION_SET(FULLAPI) ENVIRONMENT_TYPE(EXEC) SYNCONRETURN(NO) LANGUAGE_BL
DATAPGM1 OK ASKTIME ABSTIME(3675984580368) COBOLII STMT_#(00085)
DATAPGM1 OK FORMATTIME ABSTIME(3675984580368) DDMYY('27/06/16') DATESEP(X'61') TIME('02:49:40') TIMESEP(X'7A') COBOLII STMT_#(00090)
DATAPGM1 OK GET_CONTAINER('request-cont') CHANNEL('payroll') INTO(AT X'2570C190') FLENGTH(576) COBOLII STMT_#(00103)
DATAPGM1 OK RECEIVE MAP('DATAM1') INTO(AT X'2570C468') MAPSET('DATAMAP') TERMINAL COBOLII STMT_#(00271)
DATAPGM1 OK PUT_CONTAINER('request-cont') CHANNEL('payroll') FROM(AT X'2570C190') FLENGTH(576) DATATYPE(1019) COBOLII STMT_#(00410)
DATAPGM1 LINK PROGRAM('DATABUS1') CHANNEL('payroll') COBOLII STMT_#(00416)
---
DATABUS1 ASSIGN COBOLII STMT_#(00053)
s DATABUS1 OK GET_CONTAINER('request-cont') CHANNEL('payroll') INTO(AT X'25714D50') FLENGTH(576) COBOLII STMT_#(00056)
DATABUS1 OK READ FILE('PAYROLL') INTO('100001TEST QUEENSBURY HSE BRIGHTON SUSSEX 75529900&1234.') LENGTH(150) RIDFLD(AT X'A5714D54') EQUA
---
DATABUS1 *ASRA* UPDATE ABEND_RECORD ERROR_OFFSET(DA8) ABEND_TOKEN(253E4008) FAILING_PROGRAM(DATABUS1) STOKEN(4022BDB00000002F) ABEND_CODE(ASR)
---
DATABUS1 OK GETMAIN SET(X'25716CA8') FLENGTH(14904) SYSEIB ASM STMT_#(00000460)
---
DATABUS1 OK GETMAIN SET(X'2571A6F8') FLENGTH(4080) SYSEIB ASM STMT_#(00000460)
---
DATABUS1 OK GETMAIN SET(X'2571B6F8') FLENGTH(7232) SYSEIB ASM STMT_#(00000460)
---
DATABUS1 OK GETMAIN SET(X'2571D348') FLENGTH(4080) SYSEIB ASM STMT_#(00000460)
---
DATABUS1 OK ADDRESS SYSEIB ASM STMT_#(00000247)
---
DATABUS1 OK LOAD PROGRAM('CEEMENU3') SET(X'26157000') FLENGTH(54128) ENTRY(X'A6157000') SYSEIB ASM STMT_#(00000285)
---
DATABUS1 OK GETMAIN SET(X'2571E348') FLENGTH(4080) SYSEIB ASM STMT_#(00000460)
---
DATABUS1 OK WRITEQ TD QUEUE('CESE') FROM(' CP62DAT1 20160627024943 CEE3207S The system detected a data exception(System Completion Code=0C
```

Alternate example - offsets

```
C:\PROF Trace Events                                     Record 00000001 More: < >
Command ==>                                             Scroll ==> CSR
02:49:40.376108 Level 0 Gap < 000000.008 >             Format ==> SNGL
/ Time (LOCAL)
s 02:49:40.376108 AP 00E1 EIP ENTRY GET-CONTAINER REQ(0004)
_ 02:49:40.376112 AP E160 EXEC ENTRY GET CONTAINER('request-cont') CHANNEL('payroll') INTO(AT X'25714D50') FLENGTH(576) COBOLII STMT_#(00056)
_ 02:49:40.376113 AP F801 EIBAM ENTRY GET CONTAINER
_ 02:49:40.376113 PG 1700 PGCH ENTRY INQUIRE_CHANNEL CHANNEL_NAME(payroll)
_ 02:49:40.376113 PG 1701 PGCH EXIT INQUIRE_CHANNEL/OK CONTAINER_POOL_TOKEN(253EE030)
_ 02:49:40.376115 PG 1900 PGCR ENTRY GET_CONTAINER_INTO POOL_TOKEN(253EE030) CONTAINER_NAME(request-cont) CALLER(EXEC) ITEM_BUFFER(25714D50 , 0
_ 02:49:40.376117 PG 1901 PGCR EXIT GET_CONTAINER_INTO/OK USERACCESS(ANY) DATATYPE(CHAR) ITEM_BUFFER(25714D50 , 00000240 , 00000240) GENERATIO
_ 02:49:40.376117 AP F802 EIBAM EXIT GET_CONTAINER_RESP=0 RESP2=0
_ 02:49:40.376117 AP D500 UEH EVENT LINK-TO-USER-EXIT-PROGRAM GLUEEXIT AT EXIT POINT XEIOUT
_ 02:49:40.376117 AP D501 UEH EVENT RETURN-FROM-USER-EXIT-PROGRAM GLUEEXIT WITH RETURN CODE 0
_ 02:49:40.376121 AP E161 EXEC EXIT GET_CONTAINER('request-cont') CHANNEL('payroll') INTO(AT X'25714D50') FLENGTH(576) COBOLII STMT_#(00056)
_ 02:49:40.376123 AP 00E1 EIP EXIT GET-CONTAINER OK REQ(00F4)
***** Bottom of Data *****
```

```
***** Top of data *****
+0020 Code... AP 00E1
+003C STCK... D0F43267FD62C824 LSN.... 0000000000000001
Date... 2016-06-27 Monday Time... 02:49:40.376108.508

+0014 JOBN..... 'CCVT53I '

+001C TREN..... Trace entry
+0020 CALLER..... +16 POINTID.... 00E1 TYPE..... E0 TASK..... +99 KE_NUM..... 008E
+002A OWNER..... 9C6B TCB..... 'QR ' TCBADDR.... 008C6E88 RETADDR.... A5597AA2

+0044 PARMs..... Parameters
+0046 PARM..... 1
+0000 257138B0 09003414 40404040 40404040 00040200 *..... *
***** End of data *****
```

Contact Information

- Ezriel Gross' Email egross@rocketsoftware.com
- Website www.rocketsoftware.com
- Phone Office 781-684-2672
- Phone Cell 201-707-2786

Your feedback is important!

Submit a session evaluation for each session you

www.share.org/evaluation

