

## How to use CICS TS for z/VSE Trace data

### Contents

An Overview of CICS Trace .....	2
CETR Screens .....	3
CICS Trace Entries .....	5
Component .....	5
Point ID.....	5
Trace Level.....	5
Entry Type.....	6
Standard versus Special CICS Trace.....	7
Auxiliary Trace.....	7
User Trace .....	8
Formatting Trace Data.....	8
Abbreviated and Selected Full Trace Output .....	9
Abbreviated Trace Data for an EXEC CICS READ with I/O.....	21

## How to use CICS TS for z/VSE Trace data

### An Overview of CICS Trace

CICSTS was engineered to contain a very powerful and granular trace facility. It was primarily designed for use by CICS Level 2 and Level 3 support to help in diagnosing CICS problems, and a lot of the content will only make sense to a person who has access to the CICS source code and compilation listings. However, that is not to say that it is never of use to a customer or a vendor of a CICS product, especially when you consider there is an EXEC CICS interface to add User trace data.

There is one trace table and it is allocated in 31-bit Partition Getvis storage very early during CICS initialisation according to the SIT TRTABSZ size parameter. We normally advise customers to use a *minimum* of 4096 (K) because a small trace table can result in CICS Support not being able to diagnose a reported problem due to a lack of required data. The CETR transaction can be used to increase its size, subject to Getvis availability, but the size can also be decreased.

The trace table is a series of 4K blocks that contain variable-length trace entries, with the maximum size of a single entry being a little less than 4K. CICS starts writing entries at the beginning and continues until it cannot fit a complete entry at the end, when it starts at the beginning again – this is known as “wraparound mode”. When trace data is formatted, CICS always starts with the oldest entry and finishes with the newest entry, wrapping from the end to the start as required.

Unlike CICSTS for z/OS, internal trace on z/VSE does not have the same cpu overhead, and hence many customers will run with all basic trace entries enabled. I do not have reliable data for the cost, which is impossible to predict accurately for any CICS system, but it could be in the order of about a +6% cpu delta with SIT STNTR=1. If you want to know what it costs, you will need to monitor cpu usage with it enabled and then disabled. In CICSTS for z/OS I have measured a repeatable 30%+ cpu delta with the same default settings as z/VSE compared to have no trace active, and hence z/OS customers typically run with no Internal Trace active or with a subset of trace entry types active.

Internal Trace can be completely turned off but check with your performance product Vendors as this may impact their products (ASG TMON is not affected). If you send us a CICS dump with trace completely off or without entries that are typically required, there is a possibility that we may not be able to diagnose the problem.

Whatever is being captured currently in the Internal Trace table data can be written to Auxiliary Trace Datasets (DFHAUXT and DFHBUXT), which may be required to capture larger volumes of trace data in exceptional circumstances. Depending on trace data volume and dasd speed, this can increase response times by a factor of up to 6 times (and potentially more) in a production system and hence must be used carefully as directed by CICS Support.

When a Transaction Dump is produced, CICS issues a GETVIS macro for a maximum of SIT TRTRANSZ and copies the data specified by SIT TRTRANTY (normally just the failing transaction) to avoid it being lost by other trace activity while the dump is being produced.

The SIT macro, SIT overrides in the JCL PARM= and/or SYSIPT, and the CETR transaction are used to manage the trace activity. There are many permutations in CETR, and I would recommend playing with CETR in a Test CICS system to see what is possible and make the mistakes there before you need to use CETR for a production CICS problem. Using CETR introduces the concept of “Standard” and “Special” trace, but I will cover that later.

Trace settings are not affected by using SIT START=AUTO and Warm Start information.

## How to use CICS TS for z/VSE Trace data

### CETR Screens

```
CETR                      CICS Trace Control Facility          SYSA IYBTZCCA

Type in your choices.

Item                      Choice          Possible choices
Internal Trace Status    ===>  STARTED      STArted, STOpped
Internal Trace Table Size ===>  4096  K         16K - 1048576K

Auxiliary Trace Status   ===>  STOPPED      STArted, STOpped, Paused
Auxiliary Trace Dataset  ===>  A           A, B
Auxiliary Switch Status  ===>  NO          NO, NExt, All

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
```

PF4 from the primary screen. PF8 and PF7 to scroll. I changed the EI and FC component trace levels from the default value of 1.

```
CETR                      Component Trace Options          SYSA IYBTZCCA

Over-type where required and press ENTER.          PAGE 1 OF 3
Component Standard          Special
-----
AP          1          1-2
BF          1          1
BM          1          1
BR          1          1-2
CP          1          1-2
DC          1          1
DD          1          1-2
DH          1          1-2
DI          1          1
DM          1          1-2
DS          1          1-2
DU          1          1-2
EI          1-2        1-2
FC          1-2        1-2
GC          1          1-2
IC          1          1-2
IE          1          1-2

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

## How to use CICS TS for z/VSE Trace data

PF5 from the primary screen.

```
CETR                               Transaction and Terminal Trace          SYSA IYBTZCCA
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 ZCP Trace                  ===>                                ON, OFF

When finished, press ENTER.

PF1=Help      3=Quit      9=Error List
```

## How to use CICS TS for z/VSE Trace data

### CICS Trace Entries

CICS trace entries are described in the CICS TS for VSE/ESA Trace Entries manual SX33-6108 produced for release 1.1.0. The Enhancements Guide for 1.1.1, and for CICS TS for z/VSE 2.1 and 2.2 have corrections and descriptions of the trace entries that were added. We have never republished the whole CICS TS for VSE/ESA 1.1.0 library.

Some of the trace entries conforms to a pre-CICS TS release “old style” format, and are documented before the “new style” CICS TS entries.

Otherwise, the trace entry has:

- A COMPONENT (or DOMAIN)
- A Point ID
- A trace level
- An entry type
- One or more sets of output data

#### Component

If you press PF1 (Help) in CETR you will see a list of the components, for example, AP is the Application Domain, EI is the Exec Interface and FC is File Control. The CETR help facility provides a very good explanation of how to use it.

#### Point ID

When concatenated with the Component, it uniquely identifies the trace entry because the same Point ID might be used in several Components. However, for historical reasons, the “AP” Component ID can be used in the trace entry for other Components, for example, FC trace uses AP 04xx entries!

#### Trace Level

SIT STNTR sets the default trace level for all Components. Most z/VSE customers use STNTR=1 to make level 1 the default. STNTR=OFF turns Internal Trace OFF and there are other possibilities.

Level 1 trace entries are used for the typical tracing needs of a Component. Level 2 and higher provide more detailed information but increase the cpu overhead.

If you press PF4 in CETR (Components), you can set the Standard level or levels that are active for a component. For example, you could set EI 1-2 to produce basic EI trace plus the detailed content of the EXEC CICS commands. You can also use OFF, or ALL to turn on all available levels, normally when advised to do so by CICS Support. Although it is not normally used, you could set EI 2 and avoid level 1 EI trace entries being produced. PARM field or SYSIPT SIT *overrides* (but NOT SIT macro operands) can be used to change a Component trace level from the STNTR default, for example, STNTREI=(1,2).

The higher SM trace levels will avoid using Storage Management fast paths that reduce the SM overhead (EXEC CICS and internal GETMAIN and FREEMAIN requests can be expensive), but you would not normally enable them unless asked to do so by CICS Support.

## How to use CICS TS for z/VSE Trace data

You may be asked to set these trace levels or find them useful:

AP 1-2 Often asked for with IS 1-2 but may be useful in other cases.

EI 1-2 The level 2 trace data provides detailed data like that seen in CEDF output and may identify the statement number in the program source.

FC 1-2 Provides control block information used to diagnose VSAM problems and will also show RIDFLD values.

IS 1-2 Shows detailed MRO or ISC diagnostic data.

SO -12 Shows data transfer from TCPIP SERVICE socket I/O (but not EZASOCKET) for SOxxxx abends, and can be used to identify performance problems that are due to slow socket I/O requests.

WB 1-2 Shows detailed CICS Web Support diagnostic data for WBxxxx abends

### Entry Type

#### Entry

The trace entry that is produced on entry to a CICS module. Describes what the module is being asked to do.

#### Exit

The trace entry that is produced on exit from a module. Contains the result of the module execution and if it completed successfully or encountered an error.

#### Event

The CICS design says that trace data at this point in the code might be useful.

#### Exception

An error condition, which may or may not be expected and may or not be handled by CICS or an application program. For example, a program check. These are highlighted in the formatted trace with an **\*EXC\*** tag.

\*EXCU relate the User trace exception entries.

Exception entries cannot be suppressed, even if you turn CICS Internal Trace off.

#### Data

Several data values can be stored in the trace entry.

Some of these are described as the “xxxx parameter list” or “xxxxxxx\_ARG, none of which can be interpreted without access to the CICS source code.”

They may contain (part of) a CICS or another product’s control block, the address of a control block, the return code from a request, (part of) the content of an I/O buffer etc.

“Kernel error data” contains PSW and register data that relate to an abend. You will also see this in DFHPD4n0 KE=3 output. Use DFHPD410 for CICS TS 1.1.1, DFHPD420 for 2.1 and DFHPD430 for 2.2.

## How to use CICS TS for z/VSE Trace data

### Standard versus Special CICS Trace

CETR PF4 shows settings for "Standard" and "Special" trace.

Most of the time, trace data is produced according to the "Standard" settings, subject to the "Internal Trace Status" being "STARTED" and the "Master System Trace Flag" being "ON". SIT INTTR and SYSTR are used to set the values during CICS initialisation.

Special trace allows you to set different Component level options for specific transaction ids or terminal ids. The default level is set by SIT SPCTR and SIT SPCTRxx overrides can be used to define a different level for Component "xx".

A CICS task will be attached with one of these trace settings:

1. STANDARD trace settings active – the default
2. SPECIAL trace settings active
3. SUPPRESSED – no trace data (not available by terminal id)

The trace setting will remain in place until either it is changed to a different value or a COLD start is performed, which returns everything to STANDARD.

If you use CETR to set the "Master Trace Flag" to "OFF", CICS will only trace tasks that are attached with the SPECIAL trace attribute! I used this to allow me to use Auxiliary Trace on a production system when there was a problem with a specific MQ/VSE transaction id but I did not know when it would happen. By tracing only few Components as Special trace, the overhead was sufficiently low to avoid unacceptable response times.

### Auxiliary Trace

DLBLs for both DFHAUXT and DFHBUXT are normally defined.

SIT parameters AUXTR and AUXTRSW control what happens when CICS initialises.

CETR is normally used to start and stop Auxiliary Trace (the "Auxiliary Trace Status"), and to determine how the datasets are used, although CEMT can also be used.

It is possible to set the "Auxiliary Trace Dataset" to either "A" or "B" to determine which is used first before it is started. Changing the dataset while Auxiliary Trace is active will cause a switch.

"Auxiliary Switch Status" tells CICS what to do when the dataset fills:

1. NO – don't switch, stop recording data when the selected dataset is full
2. NEXT – switch to the next dataset but stop when it fills (i.e. only one switch)
3. ALL – keep switching until Auxiliary Trace is stopped (this can be useful for tracing a CICS initialisation problem with SIT AUXTR=ON and AUXTRSW=ALL)

## How to use CICS TS for z/VSE Trace data

### User Trace

EXEC CICS ENTER TRACENUM is normally used to write standard or exception User trace entries.

SIT USERTR determines whether User trace data is enabled when CICS is initialised. CETR can be used to disable or enable it. However, exception trace entries are always written.

Interpretation of User trace entries is not discussed in this document.

### Formatting Trace Data

Transaction Dump trace formatting is always performed for Transaction Dumps when running DFH DU4n0 - DFH DU410 for CICS TS 1.1.1, DFH DU420 for 2.1 and DFH DU430 for 2.2.

Auxiliary trace is formatted by DFHTU4n0, where “n” is as above. For all the options see the CICS TS for VSE/ESA 1.1.0 Operations and Utilities Guide SC33-1654 and check for any changes in the Enhancements Guide for 1.1.1, 2.1 and 2.2. In addition to the basic choice of formatting detail using ABBREV or FULL, there are many ways to select only some of the trace data. For example, you can use TASKID=(task#1<,task#2 . .>) to format data for only one or more taskids. Although it is not documented, the trace output can be written to a disk file instead of printed to SYSLST by using a DLBL DFHAXPR in the JCL.

Trace data in a CICS System Dump (an SDUMP) is formatted by DFHPD4n0, where “n” is as above, by including TR=1 (ABBREV), TR=2 (FULL) or TR=3 (both). DFHTU4n0 selection parameters can be included by using a TR=n, TRS=<...>, for example:

```
CALL DFHPD410 DATA TR=2, TRS=<TASKID=00216>
```



## How to use CICS TS for z/VSE Trace data

### Abbreviated and Selected Full Trace Output

I will use trace data from a simple Assembler program to explain how to interpret it. CETR was used to enable level 2 trace for some of the Components, and these entries will be identified. The trace was produced after the program had run successfully so that trace entries relating to loading the program and other transaction activity were mostly removed. Even so, there is a lot of output.

Abbreviated trace provides a high-level view of the CICS activity but does not always provide the data that is required, although it is relatively easy to find the equivalent Full trace entries. We start at the point where Terminal Control has received the request to run "TRA1".

```
DSTCB 1 DS 0022 DSDS3 EVENT VSE_WAIT_ENTRY                                =000110=
```

DSTCB is the CICS task id, which is the CICS Dispatcher.

You will also see: TCP for Terminal Control, XM for the Transaction Manager that is used during task initialisation and termination, plus other internal CICS system task id names. You will also see 00049, which is the TRA1 task number.

1 is the numeric CICS z/VSE subtask number. 1 is normally the QR subtask which normally does more than 90% of the Work of running transactions. 2 is typically the RO subtask which is used for functions like loading programs.

DS 0022 is the trace id. DS 0022 is produced when the CICS subtask goes into a wait state because it has no work.

<DFH>DSDS3 is the CICS module name.

The remainder shows the type of trace entry, a summary of the trace data and then a trace entry sequence number, which can be used to find the matching entry in the full trace output. If the trace is formatted with the TIMESTAMP option, you will see the time difference between the entries and the top of the page will show the time of the first entry on the page to 10 decimal places of seconds.

```
DSTCB 1 DS 0023 DSDS3 EVENT VSE_WAIT_EXIT                                *=000111=
```

The DS 0023 is when subtask number 1 comes out of the wait state. The "\*" at the start of the sequence number says that the difference in time was more than a threshold value, which is 0.0128 seconds by default. It comes out of a wait state because I asked CICS to run transaction TRA1, but when CICS is idle you will see the ICV timer effect.

## How to use CICS TS for z/VSE Trace data

TCP 1 DS 0005 DSSR EXIT WAIT\_OLDW/OK =000112=

This wakes up the TCP task and it exits the wait state it was in.

TCP 1 XM 0801 XMSR ENTRY INQUIRE\_MXT =000113=

TCP issues a request to check MXT because the status will determine if a new task can be created. This is the entry to the module that performs the request.

TCP 1 XM 0802 XMSR EXIT INQUIRE\_MXT/OK 14,1 =000114=

Then we see the matching exit trace and the response.

TCP 1 AP FD14 ZRAC EVENT RECV\_ANY 01C6F420,SFC2 =000115=  
 TCP 1 AP FCE0 ZGET ENTRY GETMAIN RPL 01C6F420,SFC2 =000116=  
 TCP 1 AP FCE1 ZGET EXIT GETMAIN/OK 015CA0C8 =000117=  
 TCP 1 AP FC90 VIO EVENT TCTTE(01C6F420) IYBTSFC2,0004,RECEIVE,DATA,OIC =000118=  
 TCP 1 SM 0C01 SMMG ENTRY GETMAIN C8,01C6F420,NO,TERMINAL =000119=  
 TCP 1 SM 0C02 SMMG EXIT GETMAIN/OK 015D10E0 =000120=  
**TCP 1 AP FD11 ZATT ENTRY ATTACH 01C6F420,SFC2 =000121=**

TCP issues the ATTACH a new CICS task 39 to run TRA1 on the terminal SFC2.

TCP 1 XM 1101 XMAT ENTRY ATTACH TRA1,0,T,NO,YES,01C6F420 , 01F80000,TERMINAL =000122=  
 TCP 1 XM 0401 XMLD ENTRY LOCATE\_AND\_LOCK\_TRANDEF TRA1,YES =000123=  
 TCP 1 DD 0301 DDLO ENTRY LOCATE 01500040,02AE16E4,TXD,TRA1 =000124=  
 TCP 1 DD 0302 DDLO EXIT LOCATE/OK 01C3B0C8 , D7000000 =000125=  
 TCP 1 XM 0402 XMLD EXIT LOCATE\_AND\_LOCK\_TRANDEF/OK 01C3A440 , 000000EB,TRA1 =000126=  
 TCP 1 DS 0002 DSAT ENTRY ATTACH 01507900,0,1,NON\_SYSTEM,01507900 , 0000039C =000127=  
 TCP 1 DS 0003 DSAT EXIT ATTACH/OK 01820009 =000128=  
 TCP 1 XM 1102 XMAT EXIT ATTACH/OK 01507900 , 0000039C,0000039C =000129=  
 TCP 1 AP FD91 ZATT EXIT ATTACH =000130=  
**TCP 1 DS 0004 DSSR ENTRY WAIT\_OLDW DFHZDSP,TCP\_NORM,00584C20,CSTP,NO,IDLE =000131=**

As TCP has nothing else to do, it enters a wait state

## How to use CICS TS for z/VSE Trace data

Transaction Manager prepares the new task for execution.

```
XM  1 DS 0012 DSKE  ENTRY TASK_REPLY          01A16080,0296D180          =000132=  
XM  1 XM 1305 XMTA  ENTRY TASK_REPLY          01507900,01820009,01820009      =000133=  
XM  1 SM 0F01 SMAR  ENTRY ALLOCATE_TRANSACTION_STG BELOW,USER,YES,NO,NO    =000134=  
XM  1 SM 0F02 SMAR  EXIT  ALLOCATE_TRANSACTION_STG/OK                       =000135=  
XM  1 PG 0801 PGXM  ENTRY INITIALIZE_TRANSACTION                                =000136=  
XM  1 PG 0802 PGXM  EXIT  INITIALIZE_TRANSACTION/OK                         =000137=  
XM  1 AP 0590 APXM  ENTRY INIT_XM_CLIENT                                           =000138=  
XM  1 SM 0301 SMGF  ENTRY GETMAIN          818,YES,00,AP_USER,TASK24      =000139=
```

A request is made for X'818' bytes of task storage for task-related control blocks.

```
XM  1 SM 0302 SMGF  EXIT  GETMAIN/OK          006C0008                      =000140=
```

The GETMAIN worked, although the allocation start address was 006C0000 because there is an 8-byte check zone at the beginning so that CICS can check for a storage violation. 006C0008 is the start address for the code to use. After rounding the request size to a multiple of X'10', there is an additional check zone at the end.

The check zones contain characters 'B0000049' to say that it is user task storage below the 16MB line for task 49.

## How to use CICS TS for z/VSE Trace data

The task starts to execute.

```
00039 1 AP EA00 TMP ENTRY LOCATE PFT,TRAN =000141=
00039 1 AP EA01 TMP EXIT LOCATE PFT,TRAN,01A1DEA0,NORMAL =000142=
00039 1 SM 0301 SMGF ENTRY GETMAIN FA0,YES,EI_TRACE,USER =000143=
00039 1 SM 0302 SMGF EXIT GETMAIN/OK 01B00008 =000144=
00039 1 AP 0591 APXM EXIT INIT_XM_CLIENT/OK =000145=
00039 1 AP 1790 TFXM ENTRY INIT_XM_CLIENT 01C6F420 , 01F80000 =000146=
00039 1 XM 1001 XMIQ ENTRY SET_TRANSACTION TERMINAL,01C6F420 =000147=
00039 1 XM 1002 XMIQ EXIT SET_TRANSACTION/OK =000148=
00039 1 AP 1791 TFXM EXIT INIT_XM_CLIENT/OK 00000000,00000000,YES,NO =000149=
00039 1 US 0401 USXM ENTRY INIT_TRANSACTION_USER 00000000,YES =000150=
00039 1 XS 0401 XSXM ENTRY ADD_TRANSACTION_SECURITY 00000000 , 00000000 =000151=
00039 1 XS 0402 XSXM EXIT ADD_TRANSACTION_SECURITY/OK =000152=
00039 1 US 0402 USXM EXIT INIT_TRANSACTION_USER/OK 0151809F , 0151A070,0 =000153=
00039 1 DS 0002 DSAT ENTRY SET_PRIORITY 1 =000154=
00039 1 DS 0003 DSAT EXIT SET_PRIORITY/OK =000155=
00039 1 AP 05A0 APRM ENTRY TRANSACTION_INITIALIZATION =000156=
00039 1 AP 05A1 APRM EXIT TRANSACTION_INITIALIZATION/OK =000157=
00039 1 AP 0590 APXM ENTRY BIND_XM_CLIENT =000158=
00039 1 AP 0591 APXM EXIT BIND_XM_CLIENT/OK =000159=
00039 1 AP 1790 TFXM ENTRY BIND_XM_CLIENT 01C6F420 , 01F80000 =000160=
00039 1 AP 1791 TFXM EXIT BIND_XM_CLIENT/OK YES,PROGA1 =000161=
00039 1 XS 0701 XSRC ENTRY CHECK_CICS_RESOURCE TRA1,TRANSATTACH,EXECUTE =000162=
00039 1 XS 0702 XSRC EXIT CHECK_CICS_RESOURCE/OK =000163=
00039 1 MN 0A01 MNXM ENTRY TRANSACTION_INITIALISATION D5B3E5924D9D704A,0000000000000000,0000000000000000,D5B3E5924DBEF758 =000164=
00039 1 MN 0A02 MNXM EXIT TRANSACTION_INITIALISATION/OK =000165=
00039 1 PG 0901 PGPG ENTRY INITIAL_LINK PROGA1 =000166=
```

CICS starts Link Level 1.

```
00039 1 LD 0001 LDLD ENTRY ACQUIRE_PROGRAM 01AF7110 =000167=
00039 1 LD 0002 LDLD EXIT ACQUIRE_PROGRAM/OK 00616000,00616000,262,0,REUSABLE,SDSA,OLD_COPY =000168=
00039 1 AP 1940 APLI ENTRY START_PROGRAM PROGA1,CEDF,FULLAPI,EXEC,NO,015BAD50,00000000 , 00000000,1,NO =000169=
```

Now the program really starts.

## How to use CICS TS for z/VSE Trace data

```
00039 1 SM 0C01 SMMG ENTRY GETMAIN          1F0,YES,00,TASK24          =000170=
00039 1 SM 0C02 SMMG EXIT  GETMAIN/OK       006C0838                    =000171=
```

The first thing that it does is to allocate DFHEISTG. It will contain 2 check zones.

TRAI now issues an EXEC CICS READ UPDATE that fails, and the start and end of this request is highlighted in red. EI level 2 entries are in blue. FC level 2 entries are in yellow. There is also an exception trace entry in green, which is always produced. Apart from the fact that we can see that it failed because the key was not found, the abbreviated trace data is difficult to interpret because there are a lot of values whose use is not obvious.

```
00039 1 AP 00E1 EIP  ENTRY READ                                0004,006C0848 .%. ,09000602 .... =000172=
00039 1 AP E160 EXEC ENTRY READ                                'FILEB ' AT X'006161C0',AT X'006C09D0',80 AT X'006C08AC',AT X'8061622D' =000173=
00039 1 AP 04F0 FCEI ENTRY PROCESS_EXEC_ARGUMENTS 006C08B0,0059D988 =000174=
00039 1 AP EA00 TMP  ENTRY LOCATE                        AFCT,FILEB =000175=
00039 1 AP EA01 TMP  EXIT  LOCATE                        AFCT,FILEB,01A3C0C0,NORMAL =000176=
00039 1 XS 0701 XSRC ENTRY CHECK_CICS_RESOURCE          FILEB,FILE,READ =000177=
00039 1 XS 0702 XSRC EXIT  CHECK_CICS_RESOURCE/OK =000178=
00039 1 AP 04E0 FCFR ENTRY READ_INTO_UPDATE           01A385D0 , 00000007,006C09D0,50,00000000,8061622D,NO,EQUAL,NO,KEY,NO =000179=
00039 1 AP 04B0 FCVS ENTRY READ_INTO_UPDATE           01A385D0 , 00000007,006C09D0,50,00000000,8061622D,01A3F030,NO,EQUAL,NO,KE =000180=
00039 1 AP 0492 FCVR CALL  VSAM                        VSAM RPL =000181=
00039 1 AP 0493 FCVR RETUR FROM                          VSAM VSAM RPL =000182=
00039 1 AP 04B7 FCVS *EXC* VSAM                        EXCEPTION VSAM RPL =000183=
00039 1 AP 0492 FCVR CALL  VSAM                        VSAM RPL =000184=
00039 1 AP 0493 FCVR RETUR FROM                          VSAM VSAM RPL =000185=
00039 1 AP 04B1 FCVS EXIT  READ_INTO_UPDATE/EXCEPTION RECORD_NOT_FOUND,0,0,0,00000000,,LENGTH_OK =000186=
00039 1 AP 04E1 FCFR EXIT  READ_INTO_UPDATE/EXCEPTION RECORD_NOT_FOUND,0,0,0,00000000,,LENGTH_OK =000187=
00039 1 AP 04F1 FCEI EXIT  PROCESS_EXEC_ARGUMENTS/EXCEPTION BAD_EIBRCODE =000188=
00039 1 AP E161 EXEC EXIT  READ                                'FILEB ' AT X'006161C0','. . . . . =000189=
00039 1 AP 00E1 EIP  EXIT  READ                                NOTFND                                00F4,00000050 ...&,000D0602 .... =000190=
```

## How to use CICS TS for z/VSE Trace data

Let's have a look at some of the full trace. In here we will see a lot more detail and even discover the key that was not found.

```
AP 00E1 EIP ENTRY READ                                REQ(0004) FIELD-A(006C0848 .%. ) FIELD-B(09000602 ....)
                                                    Register 13 (R13) save area address
TASK-00039 KE_NUM-0022 TCB-002D3000 RET-506160B8 TIME-17:00:38.9277827739 INTERVAL-00.0010578183 =000172=
                                                    R14 return address to the application program

AP E160 EXEC ENTRY READ FILE('FILEB ' AT X'006161C0') INTO( AT X'006C09D0') LENGTH(80 AT X'006C08AC') RIDFLD( AT X'8061622D')
EQUAL UPDATE NOHANDLE ASM                          Area address          Decimal 80              Just the address not the value

TASK-00039 KE_NUM-0022 TCB-002D3000 RET-80593496 TIME-17:00:38.9277860239 INTERVAL-00.0000032500 =000173=
 1-0000 003D0000 000D0061 61DC0602 F0002800 00840000 01010C00 6161C0C6 C9D3C5C2 *.....//...0....d.....//{FILEB*
    0020 40404000 02A00400 6C09D001 03030600 6C08AC00 50000404 04806162 2D * .....%.).....%...&...../.. *
 2-0000 006161DC 006161C0 006C09D0 006C08AC 8061622D * //...//{.%.}.%.../.. *
 3-0000 006C08B0 * %.. *
 4-0000 0170038F 0119049F E3D9C1F1 0000039C E2C6C3F2 00000004 00007D00 00000000 * .....TRA1....SFC2.....'..... *
    0020 00000000 00000000 00000000 00000000 00000040 40404040 40404000 00000000 * ..... *
    0040 00000000 00000000 00000000 00000000 00000000 00 * ..... *

```

If EI level 2 is active, with AP E161 being the matching exit trace.

- Data 1: Command parameters used in CICS
- Data 2: Application parameter list with addresses seen in the AP E160 formatted entry
- Data 3: Application R1 value
- Data 4: Task EIB

```
AP 04E0 FCFR ENTRY - FUNCTION(READ_INT0_UPDATE) FCTE_TOKEN(01A385D0 , 00000007) BUFFER_ADDRESS(006C09D0) BUFFER_LENGTH(50) hex
ENVIRONMENT_IDENTIFIER(00000000) RECORD_ID_ADDRESS(8061622D) GENERIC(NO) KEY_COMPARISON(EQUAL) PRIVILEGED_REQUEST
(NO) RECORD_ID_TYPE(KEY) TOKEN_REQUEST(NO)

TASK-00039 KE_NUM-0022 TCB-002D3000 RET-8169D9D0 TIME-17:00:38.9277947114 INTERVAL-00.0000011250 =000179=
 1-0000 00780000 00000038 00000000 00000000 B5AB46EC 20000000 03000100 00000000 *.....F..... *
    0020 00000000 01A385D0 00000007 00000000 006C09D0 00000050 00000000 00000000 * .....te}.....%.)...&..... *
    0040 00000000 00000000 00000000 8061622D 00000000 00000000 00000000 00000000 * ...../..... *
    0060 00000000 00000000 02020000 02010000 00000200 A5000000 * .....v... *

AP 04B0 FCVS ENTRY - FUNCTION(READ_INT0_UPDATE) FCTE_TOKEN(01A385D0 , 00000007) BUFFER_ADDRESS(006C09D0) BUFFER_LENGTH(50)
ENVIRONMENT_IDENTIFIER(00000000) RECORD_ID_ADDRESS(8061622D) WORK_ELEMENT_ADDRESS(01A3F030) GENERIC(NO)
KEY_COMPARISON(EQUAL) PRIVILEGED_REQUEST(NO) RECORD_ID_TYPE(KEY) TOKEN_REQUEST(NO)

TASK-00039 KE_NUM-0022 TCB-002D3000 RET-817556A0 TIME-17:00:38.9277975239 INTERVAL-00.0000028125 =000180=
 1-0000 00780000 00000038 00000000 00000000 B5ABC6EC 20000000 03000100 00000000 * .....F..... *
    0020 00000000 01A385D0 00000007 00000000 006C09D0 00000050 00000000 00000000 * .....te}.....%.)...&..... *
    0040 01A385D0 00000000 00000000 8061622D 00000000 01A3F030 00000000 00000000 * .te}...../.....t0..... *
    0060 00000000 00000000 02020000 02010000 00000200 A5000000 * .....v... *
 2-0000 C6C9D3C5 C2404040 * FILEB *
```

AP 04B0 if FC level 2 trace is active

## How to use CICS TS for z/VSE Trace data

AP 0492 FCVR CALL VSAM - VSAM RPL      **If FC level 2 trace is active**

```
TASK-00039 KE_NUM-0022 TCB-002D3000 RET-8193C656 TIME-17:00:38.9278012114 INTERVAL-00.0000036875      =000181=  
1-0000 0011003C 00000000 8061622D 006C09D0 00000050 00000050 00649030 00040000 *...../...%.})...&...&.....*  
0020 99000000 00000000 00000000 00000000 00000000 00000000 00000000 *r.....*  
2-0000 005A61D8 *./Q *  
3-0000 F0F0F0F1 F0F0 *000100 *
```

**Before calling VSAM.**

**Data 1: VSAM RPL. Request type is RPL+1D, with 04 = GET. OPTCD is RPL+20, with 99 = Keyed + Direct + Asynchronous + Update**

**Data 2: Address of RPL**

**Data 3: RPL RIDFLD - in this case the record key (Data 3 was added by a CICS TS 1.1.1 PTF)**

AP 0493 FCVR RETURN FROM VSAM - VSAM RPL      **If FC level 2 trace is active**

```
TASK-00039 KE_NUM-0022 TCB-002D3000 RET-8193C656 TIME-17:00:38.9278352739 INTERVAL-00.0000340625      =000182=  
1-0000 0011003C 00000000 8061622D 006C09D0 00000050 00000050 00649030 01040000 *...../...%.})...&...&.....*  
0020 99000000 00080010 00000000 00000000 00002104 00000000 00000000 *r.....*  
2-0000 005A61D8      Return code 08 (hex) *./Q *
```

**Error code 10 (hex) No Record Found - see z/VSE Messages Volume 2 VSE/VSAM Return and Error codes**

**After calling VSAM.**

**Data 1: VSAM RPL**

**Data 2: Address of RPL**

## How to use CICS TS for z/VSE Trace data

AP 04B7 FCVS \*EXC\* VSAM EXCEPTION - VSAM RPL

Always produced

```
TASK-00039 KE_NUM-0022 TCB-002D3000 RET-817556A0 TIME-17:00:38.9278358364 INTERVAL-00.0000005625 =000183=  
1-0000 00780000 00000038 00000000 00000000 B5ABC6EC 20000000 03000100 00000000 *.....F.....*  
0020 00000000 01A385D0 00000007 00000000 006C09D0 00000050 00000000 00000000 *....te}.....%}...&.....*  
0040 01A385D0 00000000 00000000 8061622D 00000000 01A3F030 00000000 00000000 *.te}...../.....t0.....*  
0060 00000000 00000000 02020100 02010000 02000200 A5000000 *.....v...*  
2-0000 0011003C 00000000 8061622D 006C09D0 00000050 00000050 00649030 01040000 *...../...%}...&...&.....*  
0020 99000000 00080010 00000000 00000000 00002104 00000000 00000000 *r.....*  
3-0000 005A61D8 *!/Q*  
4-0000 C6C9D3C5 C2404040 *FILEB*  
5-0000 E3D9C1F1 *TRAI*
```

Data 1: DFHF CFR parameter list

Data 2: RPL

Data 3: RPL address

Data 4: File name

Data 5: Transaction id

AP 0492 FCVR CALL VSAM - VSAM RPL

If FC level 2 trace is active

```
TASK-00039 KE_NUM-0022 TCB-002D3000 RET-8193D9F4 TIME-17:00:38.9278367739 INTERVAL-00.0000009375 =000184=  
1-0000 0011003C 00000000 8061622D 006C09D0 00000050 00000050 00649030 011C0000 *...../...%}...&...&.....*  
0020 91000000 00080010 00000000 00000000 00002104 00000000 00000000 *j.....*  
2-0000 005A61D8 *!/Q*
```

Request type is 1C = ENDREQ to tell VSAM to clear the Update lock because there will be no subsequent update request

AP 0493 FCVR RETURN FROM VSAM - VSAM RPL

If FC level 2 trace is active

```
TASK-00039 KE_NUM-0022 TCB-002D3000 RET-8193D9F4 TIME-17:00:38.9278368989 INTERVAL-00.0000001250 =000185=  
1-0000 0011003C 00000000 8061622D 006C09D0 00000050 00000050 00649030 011C0000 *...../...%}...&...&.....*  
0020 91000000 00000000 00000000 00000000 000000FF 00000000 00000000 *j.....*  
2-0000 005A61D8 ENDREQ completes without an error *!/Q*
```

AP 04B1 FCVS EXIT - FUNCTION(READ\_INTO\_UPDATE) RESPONSE(EXCEPTION) REASON(RECORD\_NOT\_FOUND) MAXIMUM\_RECORD\_LENGTH(0) RECORD\_LENGTH  
(0) UPDATE\_TOKEN(0) ACCMETH\_RETURN\_CODE(00000000) DUPLICATE\_KEY() LENGTH\_ERROR\_CODE(LENGTH\_OK)

```
TASK-00039 KE_NUM-0022 TCB-002D3000 RET-817556A0 TIME-17:00:38.9278373364 INTERVAL-00.0000004375 =000186=  
1-0000 00780000 00000038 00000000 00000000 B5ABC6EC 20000000 03000226 00000000 *.....F.....*  
0020 00000000 01A385D0 00000007 00000000 006C09D0 00000050 00000000 00000000 *....te}.....%}...&.....*  
0040 01A385D0 00000000 00000000 8061622D 00000000 01A3F030 00000000 00000000 *.te}...../.....t0.....*  
0060 00000000 00000000 02020100 02010000 01000200 A5000000 *.....v...*  
2-0000 C6C9D3C5 C2404040 *FILEB*
```

AP 04B1 if FC level 2 trace is active



## How to use CICS TS for z/VSE Trace data

AP 04E1 FCFR EXIT - FUNCTION(READ\_INT0\_UPDATE) RESPONSE(EXCEPTION) REASON(RECORD\_NOT\_FOUND) MAXIMUM\_RECORD\_LENGTH(0) RECORD\_LENGTH(0) UPDATE\_TOKEN(0) ACCMETH\_RETURN\_CODE(00000000) DUPLICATE\_KEY() LENGTH\_ERROR\_CODE(LENGTH\_OK)

```
TASK-00039 KE_NUM-0022 TCB-002D3000 RET-8169D9D0 TIME-17:00:38.9278377114 INTERVAL-00.0000003750 =000187=
1-0000 00780000 00000038 00000000 00000000 B5ABC6EC 20000000 03000226 00000000 *.....F.....*
0020 00000000 01A385D0 00000007 00000000 006C09D0 00000050 00000000 00000000 *....te}.....%.)...&.....*
0040 01A385D0 00000000 00000000 8061622D 00000000 01A3F030 00000000 00000000 *.te)...../.....t0.....*
0060 00000000 00000000 02020100 02010000 01000200 A5000000 *.....v.....*
```

AP 04F1 FCEI EXIT - FUNCTION(PROCESS\_EXEC\_ARGUMENTS) RESPONSE(EXCEPTION) REASON(BAD\_EIBRCODE)

```
TASK-00039 KE_NUM-0022 TCB-002D3000 RET-8059363C TIME-17:00:38.9278387739 INTERVAL-00.0000010625 =000188=
1-0000 00300000 00000036 00000000 00000000 B6000000 00000000 01000201 00000000 *.....*
0020 00000000 006C08B0 0059D988 00000012 *.....%....Rh....*
2-0000 0170038F 0119049F E3D9C1F1 0000039C E2C6C3F2 00000004 00007D06 02810000 *.....TRA1....SFC2.....'.a..*
0020 000000C6 C9D3C5C2 40404000 00000000 000000C6 C9D3C5C2 40404000 00000000 *...FILEB .....FILEB .....*
0040 00000000 00000000 00000000 0000000D 00000050 00 *.....&.....*
```

### AP 04F1 if FC level 2 trace is active

AP E161 EXEC EXIT READ FILE('FILEB ' AT X'006161C0') INTO('.....  
.....' AT X'006C09D0') LENGTH(80 AT X'006C08AC') RIDFLD( AT X'8061622D') EQUAL UPDATE RESP(13) RESP2(80)  
NOHANDLE ASM

```
TASK-00039 KE_NUM-0022 TCB-002D3000 RET-80593DD6 TIME-17:00:38.9278403364 INTERVAL-00.0000015625 =000189=
1-0000 008D0000 000D0061 61DC0602 F0002800 00840000 01010C00 6161C0C6 C9D3C5C2 *.....//...0....d.....//{FILEB*
0020 40404000 02A05400 6C09D000 00000000 00000000 00000000 00000000 00000000 * .....%.).....*
0040 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0060 00000000 00000000 00000000 00000000 00000000 00000000 00000001 03030600 *.....*
0080 6C08AC00 50000404 04806162 2D *%...&...../..*
2-0000 006161DC 006161C0 006C09D0 006C08AC 8061622D *...//.../{%.}..%.../..*
3-0000 006C08B0 *%.*..*
4-0000 0170038F 0119049F E3D9C1F1 0000039C E2C6C3F2 00000004 00007D06 02810000 *.....TRA1....SFC2.....'.a..*
0020 000000C6 C9D3C5C2 40404000 00000000 000000C6 C9D3C5C2 40404000 00000000 *...FILEB .....FILEB .....*
0040 00000000 00000000 00000000 0000000D 00000050 00 *.....&.....*
```

### The AP E161 is produced if EI level 2 is active, and the data values are as for AP E160.

AP 00E1 EIP EXIT READ NOTFND REQ(00F4) FIELD-A(00000050 ...&) FIELD-B(000D0602 ....)

```
TASK-00039 KE_NUM-0022 TCB-002D3000 RET-506160B8 TIME-17:00:38.9278406489 INTERVAL-00.0000003125 =000190=
```

## How to use CICS TS for z/VSE Trace data

Now the application issues a message and terminates by an EXEC CICS RETURN.

```

00039 1 AP 00E1 EIP  ENTRY SEND-TEXT                0004,006C0848 .%. ,09001806 .... =000191=
00039 1 AP E160 EXEC  ENTRY SEND                    TEXT 'TRANSACTION TRAl COMPLETE      ' AT X'0061620F',30 AT X'806161DA',TE =000192=
00039 1 SM 0C01 SMMG  ENTRY GETMAIN                  22,YES,00,CICS24_SAA                    =000193=
00039 1 SM 0C02 SMMG  EXIT  GETMAIN/OK                005A0008                                =000194=
00039 1 SM 0D01 SMMF  ENTRY FREEMAIN                  015D10E0,01C6F420                        =000195=
00039 1 SM 0D02 SMMF  EXIT  FREEMAIN/OK              TERMINAL storage at 015D10E0            =000196=
00039 1 AP 00FA BMS   ENTRY SEND-OUT                  CTRL                                     0003,00000800 ....,04000020 .... =000197=
00039 1 SM 0301 SMGF  ENTRY GETMAIN                  464,YES,00,MCPOSPA,CICS                  =000198=
00039 1 SM 0302 SMGF  EXIT  GETMAIN/OK                01A66008                                =000199=
00039 1 PG 0500 PGIS  ENTRY INQUIRE_CURRENT_PROGRAM                                     =000200=
00039 1 PG 0501 PGIS  EXIT  INQUIRE_CURRENT_PROGRAM/OK ASSEMBLER                               =000201=
00039 1 AP 00FA BMS   EVENT RLR-ENTRY                  0104,01A66008 .w-. ,01C6F420 .F4.      =000202=
00039 1 SM 0301 SMGF  ENTRY GETMAIN                  D4,YES,00,TTP,CICS                       =000203=
00039 1 SM 0302 SMGF  EXIT  GETMAIN/OK                01A66488                                =000204=
00039 1 AP 00FA BMS   EVENT RLR-EXIT                  4104,01A66008 .w-. ,01A66488 .w.h     =000205=
00039 1 AP 00FA BMS   EVENT PBP-ENTRY                  0204,01A66488 .w.h,005A0008 .!..      =000206=
00039 1 SM 0301 SMGF  ENTRY GETMAIN                  152,YES,00,MAPCOPY,CICS                  =000207=
00039 1 SM 0302 SMGF  EXIT  GETMAIN/OK                01A66578                                =000208=
00039 1 AP 00FA BMS   EVENT M32-ENTRY                  0304,01A66488 .w.h,005A0008 .!..      =000209=
00039 1 SM 0C01 SMMG  ENTRY GETMAIN                  3A,01C6F420,YES,00,TERMINAL              =000210=
00039 1 SM 0C02 SMMG  EXIT  GETMAIN/OK                015D10E0                                =000211=
00039 1 SM 0301 SMGF  ENTRY GETMAIN                  74,YES,00,MCA,CICS                       =000212=
00039 1 SM 0302 SMGF  EXIT  GETMAIN/OK                01A666E8                                =000213=
00039 1 SM 0301 SMGF  ENTRY FREEMAIN                  01A66578,MAPCOPY,CICS                   =000214=
00039 1 SM 0302 SMGF  EXIT  FREEMAIN/OK              01A666E8,MCA,CICS                       =000215=
00039 1 SM 0301 SMGF  ENTRY FREEMAIN                  01A666E8,MCA,CICS                       =000216=
00039 1 SM 0302 SMGF  EXIT  FREEMAIN/OK              01A666E8,MCA,CICS                       =000217=
00039 1 AP 00FA BMS   EVENT M32-EXIT                  4304,01A66488 .w.h,005A0008 .!..      =000218=
00039 1 AP 00FA BMS   EVENT TPP-ENTRY                  0604,01A66488 .w.h,0000001A ....      =000219=
00039 1 AP FD01 ZARQ  ENTRY APPL_REQ                  01C6F420,WRITE                           =000220=
00039 1 AP FD81 ZARQ  EXIT  APPL_REQ                  01C6F420,WRITE                           =000221=
00039 1 AP 00FA BMS   EVENT TPP-EXIT                  4604,01A66488 .w.h,00000000 ....      =000222=
00039 1 AP 00FA BMS   EVENT PBP-EXIT                  4204,01A66488 .w.h,005A0008 .!..      =000223=
00039 1 SM 0D01 SMMF  ENTRY FREEMAIN                  005A0008,005A0008                        =000224=
00039 1 SM 0D02 SMMF  EXIT  FREEMAIN/OK              CICS24 storage at 005A0008              =000225=
00039 1 AP 00FA BMS   EXIT                               0005,00000000 ....,00000000 ....      =000226=
00039 1 AP E161 EXEC  EXIT  SEND                    TEXT 'TRANSACTION TRAl COMPLETE      ' AT X'0061620F',30 AT X'806161DA',TE =000227=
00039 1 AP 00E1 EIP  EXIT  SEND-TEXT                OK                                         00F4,00000000 ....,00001806 .... =000228=

00039 1 AP 00E1 EIP  ENTRY RETURN                    0004,006C0848 .%. ,09000E08 ....      =000229=
00039 1 AP E160 EXEC  ENTRY RETURN                  ASM                                       =000230=
00039 1 SM 0301 SMGF  ENTRY FREEMAIN                  006C0838,TASK                             =000231=
00039 1 SM 0302 SMGF  EXIT  FREEMAIN/OK              006C0838,TASK                             =000232=

```

## How to use CICS TS for z/VSE Trace data

### PROGAL exits

```

00039 1 AP 1941 APLI EXIT START_PROGRAM/OK      ....,NO,PROGAL      =000233=
00039 1 LD 0001 LDLD ENTRY RELEASE_PROGRAM      01AF7110,00616000    =000234=
00039 1 LD 0002 LDLD EXIT RELEASE_PROGRAM/OK    00616000,262,SDSA   =000235=

```

### LINK Level 1 terminates

```

00039 1 PG 0902 PGPB EXIT INITIAL_LINK/OK      =000236=

00039 1 AP 05A0 APRM ENTRY TRANSACTION_TERMINATION NORMAL =000237=

```

### Syncpoint starts.

```

00039 1 AP D802 SPP ENTRY SYSTEM                =000238=
00039 1 AP D800 SPZ ENTRY INIT                  TASK-DETACH         =000239=
00039 1 AP D801 SPZ EXIT NORMAL                 =000240=
00039 1 AP 04F8 EFRM ENTRY PREPARE              1A40030,TERMINATE   =000241=
00039 1 AP 04F9 EFRM EXIT PREPARE/OK           =000242=
00039 1 AP 04F3 FCRM ENTRY PREPARE              1A3E030,TERMINATE   =000243=
00039 1 AP 04F4 FCRM EXIT PREPARE/OK           =000244=
00039 1 AP 04F8 EFRM ENTRY COMMIT              1A40030,TERMINATE   =000245=
00039 1 AP 04F9 EFRM EXIT COMMIT/OK           =000246=
00039 1 AP D800 SPZ ENTRY TERM                  TASK-DETACH         =000247=
00039 1 AP D801 SPZ EXIT NORMAL                 =000248=
00039 1 AP D803 SPP EXIT NORMAL

```

### Syncpoint ends. Then CICS starts to clean up the task environment.

```

=000249=
00039 1 AP 05A1 APRM EXIT TRANSACTION_TERMINATION/OK .... =000250=
00039 1 AP 1710 TFRF ENTRY RELEASE_FACILITY      NO,NORMAL,01C6F420,SFC2 =000251=
00039 1 AP FD0B ZISP ENTRY FACILITY_REQ         01C6F420,FREE_DETACH   =000252=
00039 1 AP FD03 ZDET ENTRY DETACH               01C6F420,SFC2         =000253=
00039 1 AP FD18 ZSDS ENTRY SEND_DFSYN           01C6F420,SFC2         =000254=
00039 1 AP FC90 VIO EVENT TCTTE(01C6F420)      IYBTSFC2,0006,SEND,DATA,0,RQE1,OIC =000255=
00039 1 SM 0E01 SMMC2 ENTRY FREEMAIN_ALL_TERMINAL 01C6F420 =000256=
00039 1 SM 0E05 SMMC2 EVENT Storage_released    TERMINAL storage at 015D10E0 =000257=
00039 1 SM 0E02 SMMC2 EXIT FREEMAIN_ALL_TERMINAL/OK =000258=
00039 1 AP FD15 ZRST ENTRY RESETSR             01C6F420,SFC2,ZSDX    =000259=
00039 1 XM 1001 XMIQ ENTRY SET_TRANSACTION      NONE,00000000        =000260=
00039 1 XM 1002 XMIQ EXIT SET_TRANSACTION/OK    =000261=
00039 1 AP FD8B ZISP EXIT FACILITY_REQ         =000262=
00039 1 AP 1711 TFRF EXIT RELEASE_FACILITY/OK   SFC2                  =000263=
00039 1 AP 0590 APXM ENTRY RELEASE_XM_CLIENT    NORMAL                =000264=

```

## How to use CICS TS for z/VSE Trace data

The Transaction Manager (XM) then does the rest.

```
XM 1 AP 0591 APXM EXIT RELEASE_XM_CLIENT/OK =000265=  
XM 1 US 0401 USXM ENTRY END_TRANSACTION =000266=  
XM 1 XS 0401 XSXM ENTRY END_TRANSACTION =000267=  
XM 1 XS 0402 XSXM EXIT END_TRANSACTION/OK =000268=  
XM 1 US 0402 USXM EXIT END_TRANSACTION/OK =000269=  
XM 1 PG 0801 PGXM ENTRY TERMINATE_TRANSACTION =000270=  
XM 1 PG 0802 PGXM EXIT TERMINATE_TRANSACTION/OK =000271=  
XM 1 SM 0F01 SMAR ENTRY RELEASE_TRANSACTION_STG =000272=  
XM 1 SM 0F0D SMAR EVENT Storage_released USER24 storage at 006C0008 =000273=  
XM 1 SM 0F0D SMAR EVENT Storage_released CICS storage at 01A66488 =000274=  
XM 1 SM 0F0D SMAR EVENT Storage_released CICS storage at 01A66008 =000275=  
XM 1 SM 0F0D SMAR EVENT Storage_released USER storage at 01B00008 =000276=  
XM 1 SM 0F02 SMAR EXIT RELEASE_TRANSACTION_STG/OK =000277=  
XM 1 MN 0A01 MNXM ENTRY TRANSACTION_TERMINATION =000278=  
XM 1 MN 0A02 MNXM EXIT TRANSACTION_TERMINATION/OK =000279=  
XM 1 XM 1306 XMTA EXIT TRANSACTION_TERMINATION/OK =000280=  
XM 1 DS 0013 DSKE EXIT TASK_REPLY/OK =000281=
```

Now the task has fully terminated.

## How to use CICS TS for z/VSE Trace data

### Abbreviated Trace Data for an EXEC CICS READ with I/O

```
00057 1 AP 00E1 EIP ENTRY READ 0004,019D44E8 ...Y,0D000602 .... =000061=
00057 1 AP E160 EXEC ENTRY READ 'FILEA ' AT X'019D4890',AT X'019D4744',AT X'819D473E',EQUAL,NOHANDLE,CO =000062=
00057 1 AP 04F0 FCEI ENTRY PROCESS_EXEC_ARGUMENTS 019D4660,00595988 =000063=
00057 1 AP EA00 TMP ENTRY LOCATE AFCT,FILEA =000064=
00057 1 AP EA01 TMP EXIT LOCATE AFCT,FILEA,01B49000,NORMAL =000065=
00057 1 AP 04E0 FCFR ENTRY READ_SET 01B40030 , 00000001,00000000,819D473E,NO,EQUAL,NO,KEY,31,USER =000066=
00057 1 AP 04B0 FCVS ENTRY READ_SET 01B40030 , 00000001,00000000,819D473E,01B4C030,NO,EQUAL,NO,KEY,31,USER =000067=
00057 1 SM 0301 SMGF ENTRY GETMAIN 013F0AAC , 00000084,CC,YES,VSWA =000068=
00057 1 SM 0302 SMGF EXIT GETMAIN/OK 005A21D0 =000069=
00057 1 SM 0301 SMGF ENTRY GETMAIN 50,YES,FCSS SET,USER =000070=
00057 1 SM 0302 SMGF EXIT GETMAIN/OK 019D6628 =000071=
00057 1 AP 0492 FCVR CALL VSAM VSAM RPL =000072=
00057 1 AP 04DA FCVR ENTER EXCPAD =000073=
00057 1 DS 0004 DSSR ENTRY WAIT_OLDW FILEA,FCIOWAIT,01328541,NO,IO =000074=
00057 1 DS 0005 DSSR EXIT WAIT_OLDW/OK =000075=
00057 1 AP 04DB FCVR EXIT EXCPAD =000076=
00057 1 AP 04DA FCVR ENTER EXCPAD =000077=
00057 1 DS 0004 DSSR ENTRY WAIT_OLDW FILEA,FCIOWAIT,01328541,NO,IO =000078=
00057 1 DS 0005 DSSR EXIT WAIT_OLDW/OK =000079=
00057 1 AP 04DB FCVR EXIT EXCPAD =000080=
00057 1 AP 04DA FCVR ENTER EXCPAD =000081=
00057 1 DS 0004 DSSR ENTRY WAIT_OLDW FILEA,FCIOWAIT,01328541,NO,IO =000082=
00057 1 DS 0005 DSSR EXIT WAIT_OLDW/OK =000083=
00057 1 AP 04DB FCVR EXIT EXCPAD =000084=
00057 1 AP 0493 FCVR RETUR FROM VSAM VSAM RPL =000085=
00057 1 AP 04B1 FCVS EXIT READ_SET/OK 50,019D6628,50,00000000, =000086=
00057 1 AP 04E1 FCFR EXIT READ_SET/OK 50,019D6628,50,00000000, =000087=
00057 1 AP 04F1 FCEI EXIT PROCESS_EXEC_ARGUMENTS/OK =000088=
00057 1 AP E161 EXEC EXIT READ 'FILEA ' AT X'019D4890',X'019D6628' AT X'019D4744',AT X'819D473E',EQUAL =000089=
00057 1 AP 00E1 EIP EXIT READ OK 00F4,00000000 .....,00000602 .... =000090=
```

The READ request has resulted in 3 VSAM I/O requests, and for each request the task has been suspended with an FCIOWAIT while waiting for I/O completion - FC=1-2 was active so you get to see more detail than with the default FC=1.