



IBM Software Group – TXSeries for Multiplatforms

TXSeries (CICS) Problem Determination

Iain Boyle, Software Group Services

iain_boyle@uk.ibm.com



ON DEMAND BUSINESS™

© 2005 IBM Corporation

Introduction

- This presentation provides some introductory details on how to perform problem determination in a TXSeries CICS environment. It includes application and system problems and subjects such as the location of useful files produced by CICS when a problem occurs, their format and some examples of real problems.
- Most of the information contained in this presentation is documented in the following manual:-
 - ▶ TXSeries CICS Problem Determination Guide (SC09-4465)



Contents

- **Sources of Information**
- **Understanding Product Logs**
- **CICS Product Dump and Trace**
- **CICS Supplied Transactions for Debugging**
- **Handling Exceptions in CICS**
- **Sample Problem Cases and Problem Determination**
- **Supplying IBM Support with Information**



Sources of Information

The following directories contain CICS diagnostic information:

- Region directory
 - ▶ /var/cics_regions/<region_name> (on Open Systems)
 - ▶ <install_drive>\var\cics_regions\<region_name> (on Windows)

- Region Data directory
 - ▶ /var/cics_regions/<region_name>/data (on Open Systems)
 - ▶ <install_drive>\var\cics_regions\<region_name>\data (on Windows)

- Region 'dumps/dir1' directory
 - ▶ /var/cics_regions/<region_name>/dumps/dir1 (on Open Systems)
 - ▶ <install_drive>\var\cics_regions\<region_name>\dumps\dir1 (on Windows)



Sources of Information – Region Directory

Diagnostic files under Region directory :

- **console.<nnnnnn>**

- ▶ This is the primary log file which logs the activities running in the region.
- ▶ Logs information related to CICS region startup, shutdown, region error/failure messages and some transaction related messages.
- ▶ All messages to stderr are redirected to this file
- ▶ Maximum size of the console is limited by MaxConsoleSize in the RD stanza. Default is 0 (unlimited)
- ▶ A new console is opened every time region is started

- **symrecs.<nnnnnn>** (Symptom Records)

- ▶ This is the primary log file which logs symptom records or problem causes.
- ▶ Logs information related to CICS Error conditions, symptom records and stack trace.



Sources of Information – Region Directory

Diagnostic files under Region directory :

- **console.msg**
 - ▶ All messages to stdout are redirected to this file

- **<region name>.env**
 - ▶ Log the environment under which the CICS subsystem started.



Sources of Information – Data Directory

Diagnostic files under Region '/data' directory

- **CSMT.out**

- ▶ Is a TDQ containing CICS messages about transactions or system errors.
- ▶ Applications can also write to CSMT using the EXEC CICS api.

- **CCIN.out**

- ▶ Is a TDQ that contains messages about client connections.

- **statsfile**

- ▶ Contains CICS region statistics output. This needs to be formatted through **cicssfnt** utility.



Sources of Information – Dumps Directory

Diagnostic files under Region 'dumps/dir1' directory

- Region 'dumps/dir1' directory is the current working directory for all the CICS processes. Hence any files generated or create by third-party software running under CICS will be found in this dumps/dir1 directory.

- **cics<pid>.traceback**
 - ▶ Logs function stack information when an illegal exception (SigILL, SigSEGV) occurs in an application or CICS internal code.

- **SYSA<nnnn>.dmp<mm>**
 - ▶ System dump file contains the region information generated on the abnormal termination of region.

- **ASRA<nnnn>.dmp<mm>**
 - ▶ Transaction dump file contains the transaction related information generated when the transaction abends.

- **core<pid>.<timestamp>**
 - ▶ Generated by the operating system when a CICS process terminates abnormally. CICS would attempt to format the generated core file, if CICS succeeds in formatting CICS would create a file with the same name as core file and suffix of .fmt will be added. The formatted file would contain the process/thread stack.



Contents

- Sources of Information
- **Understanding Product Logs**
- CICS Product Dump and Trace
- CICS Supplied Transactions for Debugging
- Handling Exceptions in CICS
- Sample Problem Cases and Problem Determination
- Supplying IBM Support with Information



Format of console.nnnnnn

```
ERZ074081I/7442 06/22/05 17:34:27.004509190 ABCTST 70246/0001  
: Starting region with attribute override 'StartType=cold'
```

- ERZ074081I/7442
 - ▶ Message ID / Sequence Number
- ERZ[xxx][yyy][I|W|E]/[zzzz]
 - ▶ xxx is the product module
 - ▶ yyy is the sub-component product module
 - ▶ I / W / E – Information / Warning / Error message
 - ▶ zzzz is the message sequence number
- 06/22/05 17:34:27.004509190
 - ▶ Date and Time stamp when the activity occurred
- ABCTST
 - ▶ Region name where the activity occurred
- 70246/0001
 - ▶ Process ID/Thread ID
- “Starting region with attribute override 'StartType=cold'.”
 - ▶ The Message describing the activity



Sample of console.nnnnnn

```

Vi for Windows - [D:\opt\var\cics_regions\IAIN\console.000000]
File Edit Macro Options Help
ERZ010190I/0020 17/12/2004 11:16:05.237259000 IAIN 3888/0001 : CICS Release Number 5.1.0.0 Revision Level s510-L040720-I
ERZ010135I/0362 17/12/2004 11:16:05.237259000 IAIN 3888/0001 : CICS region 'IAIN' is being started with locale categories 'En_US'
ERZ074081I/7442 17/12/2004 11:16:05.337403000 IAIN 3888/0001 : Starting region with attribute override 'StartType=cold'.
ERZ010151W/0411 17/12/2004 11:16:05.387475000 IAIN 3888/0001 : Cold start forced as previous cold start did not complete
ERZ010117I/0150 17/12/2004 11:16:05.407504000 IAIN 3888/0001 : CICS is performing a cold start
ERZ048042I/0107 17/12/2004 11:16:05.467590000 IAIN 3888/0001 : Region Pool initialized at address: 70000004
ERZ048043I/0108 17/12/2004 11:16:05.467590000 IAIN 3888/0001 : Task Shared Pool initialized at address: 79c00000
ERZ052025I/1710 17/12/2004 11:16:05.627820000 IAIN 3888/0001 : Dumping initialized with these options set to YES 'PCDump ABDump '.
ERZ049029I/0805 17/12/2004 11:16:05.697921000 IAIN 3888/0001 : Trace initialization has begun
ERZ049011I/0815 17/12/2004 11:16:05.747993000 IAIN 3888/0001 : Trace parameter 'TraceDirectorySystem' has been set to 'D:\opt\var\ci
cs_regions\IAIN\dumps\dir1'
ERZ049011I/0817 17/12/2004 11:16:05.788051000 IAIN 3888/0001 : Trace parameter 'TraceDirectoryUser' has been set to 'D:\opt\var\cics
tmp'
ERZ049011I/0818 17/12/2004 11:16:05.868166000 IAIN 3888/0001 : Trace parameter 'TraceFlagAux' has been set to 'OFF'
ERZ049011I/0825 17/12/2004 11:16:05.918238000 IAIN 3888/0001 : Trace parameter 'TraceFlagBuffer' has been set to 'ON'
ERZ049011I/0827 17/12/2004 11:16:05.958296000 IAIN 3888/0001 : Trace parameter 'TraceFlagExec' has been set to 'OFF'
ERZ049011I/0830 17/12/2004 11:16:06.008368000 IAIN 3888/0001 : Trace parameter 'TraceFlagExternal' has been set to 'OFF'
ERZ049011I/0833 17/12/2004 11:16:06.048425000 IAIN 3888/0001 : Trace parameter 'TraceFlagMaster' has been set to 'OFF'
ERZ049011I/0836 17/12/2004 11:16:06.088483000 IAIN 3888/0001 : Trace parameter 'TraceFlagSystem' has been set to 'OFF'
ERZ049011I/0839 17/12/2004 11:16:06.138555000 IAIN 3888/0001 : Trace parameter 'TraceFlagUser' has been set to 'OFF'
ERZ049011I/0842 17/12/2004 11:16:06.178612000 IAIN 3888/0001 : Trace parameter 'TraceMaxSizeAux' has been set to '0'
ERZ049011I/0843 17/12/2004 11:16:06.218670000 IAIN 3888/0001 : Trace parameter 'TraceMaxSizeBuffer' has been set to '131072'
ERZ049011I/0845 17/12/2004 11:16:06.258728000 IAIN 3888/0001 : Trace parameter 'TraceSystemSpec' has been set to 'all=4'
ERZ049032I/0808 17/12/2004 11:16:06.298785000 IAIN 3888/0001 : CICS is using shared memory ID '1' for trace
ERZ049033I/0811 17/12/2004 11:16:06.378900000 IAIN 3888/0001 : Trace initialization has completed
ERZ034108I/0722 17/12/2004 11:16:06.469030000 IAIN 3888/0001 : Loading region runtime database ...
ERZ069020I/6935 17/12/2004 11:16:06.989779000 IAIN 3888/0001 : ... Communications Definitions loaded
ERZ070020I/7035 17/12/2004 11:16:07.059880000 IAIN 3888/0001 : ... File Definitions loaded
ERZ073020I/7335 17/12/2004 11:16:07.109952000 IAIN 3888/0001 : ... Program Definitions loaded
ERZ078020I/7835 17/12/2004 11:16:07.150009000 IAIN 3888/0001 : ... Temporary Storage Definitions loaded
ERZ076020I/7635 17/12/2004 11:16:07.260168000 IAIN 3888/0001 : ... Terminal Definitions loaded

```

105:1:1 31x140

Command



Format of symrecs.nnnnnn

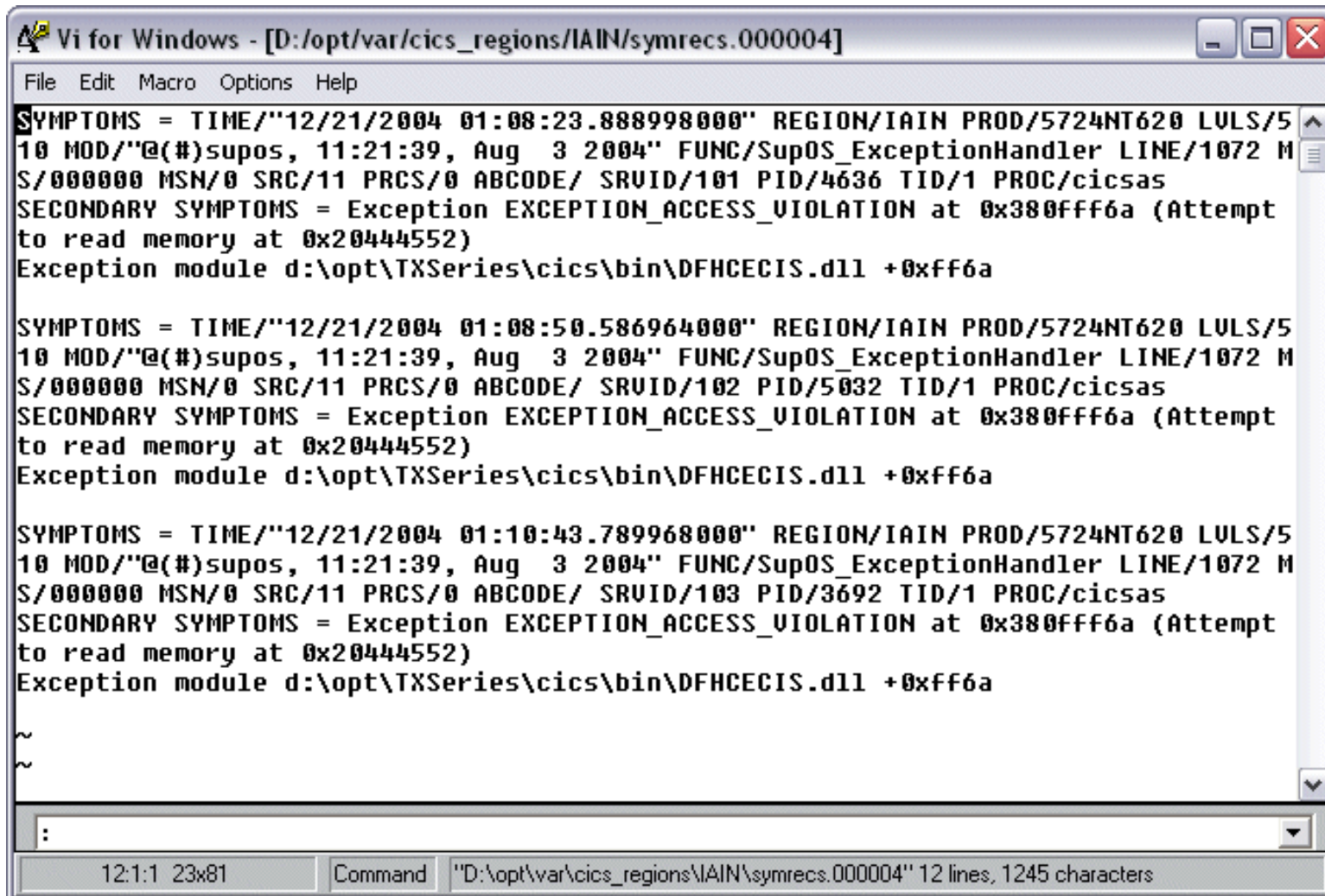
```
SYMPTOMS = TIME/"6/21/2005 11:23:37.670491000" REGION/IAIN PROD/5724NT620 LVLS/510  
MOD/"@(#)damfs, 13:55:26, Mar 24 2005" FUNC/DamFS_GetHandleByName LINE/756 MS/000000 MSN/9  
SRC/4 PRCS/1997905970 ABCODE/ SRVID/102 PID/5892 TID/1 PROC/cicsas
```

```
SECONDARY SYMPTOMS = SFS error ENC-sfs-0050: File by this name does not exist. mapped onto  
IOERR. BaseName 'FAACTIEM', FileServer '././cics/sfs/BOYLE', IndexName 'BASEINDEX'.
```

- The primary symptom data contains the following information:
 - ▶ TIME – When the condition is detected
 - ▶ REGION – The region name
 - ▶ PROD – The product component ID
 - ▶ LVLS – The product level
 - ▶ MOD – The module name and module-build timestamp
 - ▶ FUNC, LINE, and MSN – Where in the code it happens
 - ▶ ABCODE – The abend code
 - ▶ SRVID – The CICS Server ID
 - ▶ PID/TID – The Process ID and Thread ID
 - ▶ PROC – The process name
- The secondary symptom data is a text record, which might contain a message about the problem or some related data. Sometimes, this record is left blank.



Sample of symrecs.nnnnnn



```
Vi for Windows - [D:\opt\var\cics_regions\IAIN\symrecs.000004]
File Edit Macro Options Help
SYMPTOMS = TIME/"12/21/2004 01:08:23.888998000" REGION/IAIN PROD/5724NT620 LULS/5
10 MOD/"@(#)supos, 11:21:39, Aug 3 2004" FUNC/SupOS_ExceptionHandler LINE/1072 M
S/000000 MSN/0 SRC/11 PRCS/0 ABCODE/ SRUID/101 PID/4636 TID/1 PROC/cicsas
SECONDARY SYMPTOMS = Exception EXCEPTION_ACCESS_VIOLATION at 0x380fff6a (Attempt
to read memory at 0x20444552)
Exception module d:\opt\TXSeries\cics\bin\DFHCECIS.dll +0xff6a

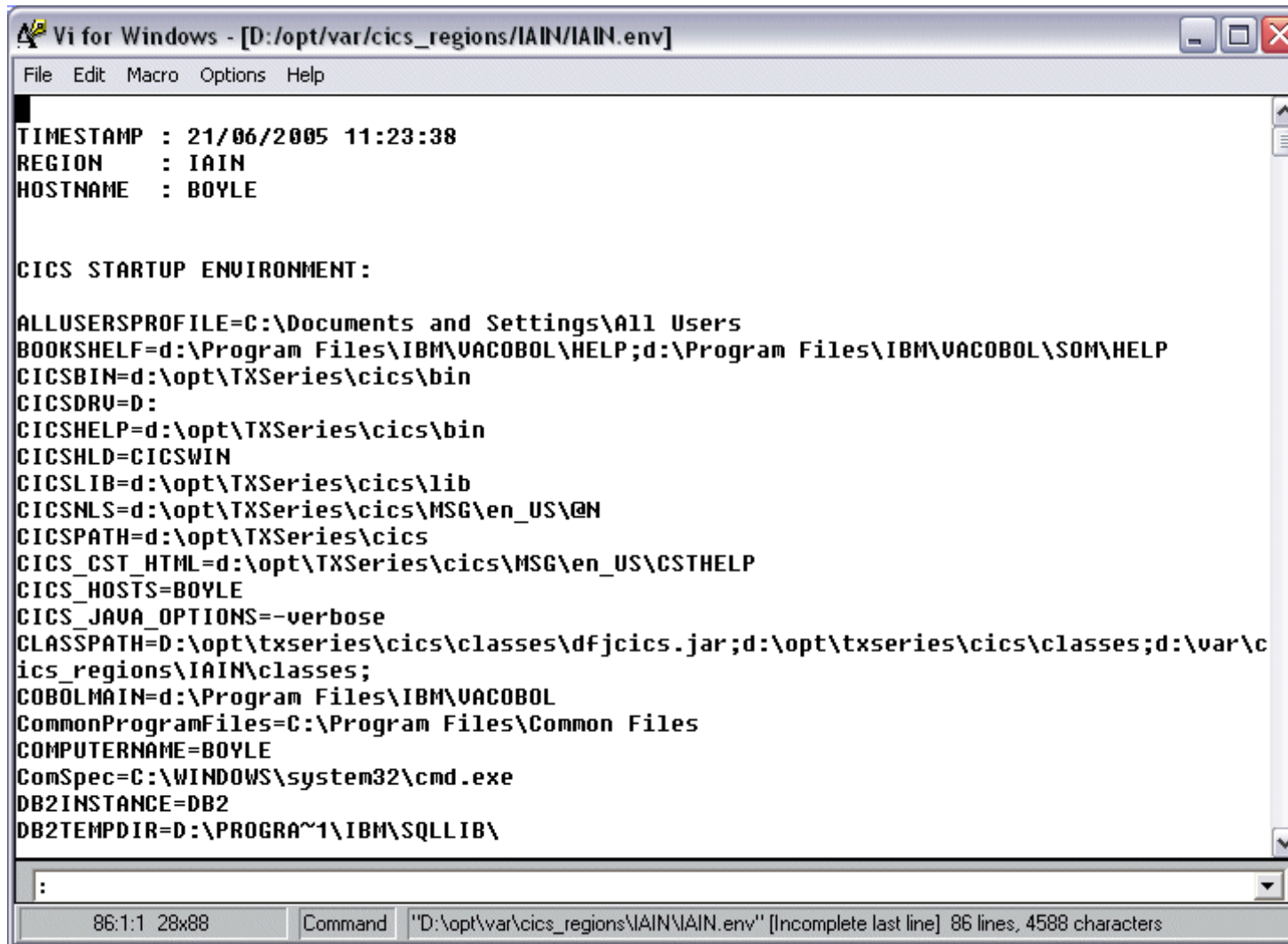
SYMPTOMS = TIME/"12/21/2004 01:08:50.586964000" REGION/IAIN PROD/5724NT620 LULS/5
10 MOD/"@(#)supos, 11:21:39, Aug 3 2004" FUNC/SupOS_ExceptionHandler LINE/1072 M
S/000000 MSN/0 SRC/11 PRCS/0 ABCODE/ SRUID/102 PID/5032 TID/1 PROC/cicsas
SECONDARY SYMPTOMS = Exception EXCEPTION_ACCESS_VIOLATION at 0x380fff6a (Attempt
to read memory at 0x20444552)
Exception module d:\opt\TXSeries\cics\bin\DFHCECIS.dll +0xff6a

SYMPTOMS = TIME/"12/21/2004 01:10:43.789968000" REGION/IAIN PROD/5724NT620 LULS/5
10 MOD/"@(#)supos, 11:21:39, Aug 3 2004" FUNC/SupOS_ExceptionHandler LINE/1072 M
S/000000 MSN/0 SRC/11 PRCS/0 ABCODE/ SRUID/103 PID/3692 TID/1 PROC/cicsas
SECONDARY SYMPTOMS = Exception EXCEPTION_ACCESS_VIOLATION at 0x380fff6a (Attempt
to read memory at 0x20444552)
Exception module d:\opt\TXSeries\cics\bin\DFHCECIS.dll +0xff6a

~
~
:
```

12:1:1 23x81 Command "D:\opt\var\cics_regions\IAIN\symrecs.000004" 12 lines, 1245 characters

Sample of <regionname>.env



```
Vi for Windows - [D:\opt\var\cics_regions\IAIN\IAIN.env]
File Edit Macro Options Help

TIMESTAMP : 21/06/2005 11:23:38
REGION    : IAIN
HOSTNAME  : BOYLE

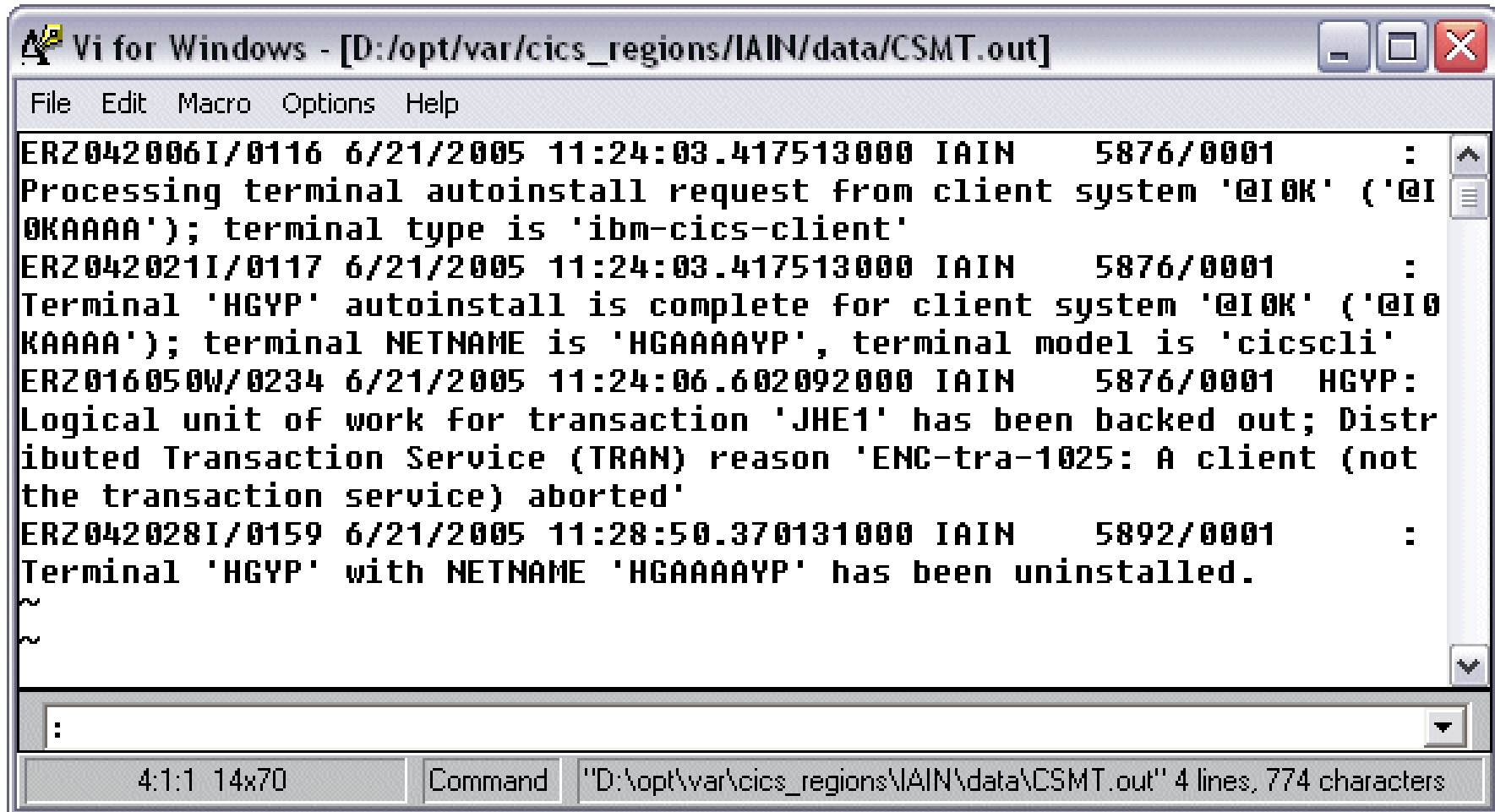
CICS STARTUP ENVIRONMENT:

ALLUSERSPROFILE=C:\Documents and Settings\All Users
BOOKSHELF=d:\Program Files\IBM\UACOBOL\HELP;d:\Program Files\IBM\UACOBOL\SOM\HELP
CICSBIN=d:\opt\TXSeries\cics\bin
CICSDRU=D:
CICSHelp=d:\opt\TXSeries\cics\bin
CICSHLD=CICSWIN
CICSLIB=d:\opt\TXSeries\cics\lib
CICSNLS=d:\opt\TXSeries\cics\MSG\en_US\@N
CICSPATH=d:\opt\TXSeries\cics
CICS_CST_HTML=d:\opt\TXSeries\cics\MSG\en_US\CSTHELP
CICS_HOSTS=BOYLE
CICS_JAVA_OPTIONS=-verbose
CLASSPATH=D:\opt\txseries\cics\classes\dfjcics.jar;d:\opt\txseries\cics\classes;d:\var\cics_regions\IAIN\classes;
COBOLMAIN=d:\Program Files\IBM\UACOBOL
CommonProgramFiles=C:\Program Files\Common Files
COMPUTERNAME=BOYLE
ComSpec=C:\WINDOWS\system32\cmd.exe
DB2INSTANCE=DB2
DB2TEMPDIR=D:\PROGRA~1\IBM\SQLLIB\

:
```

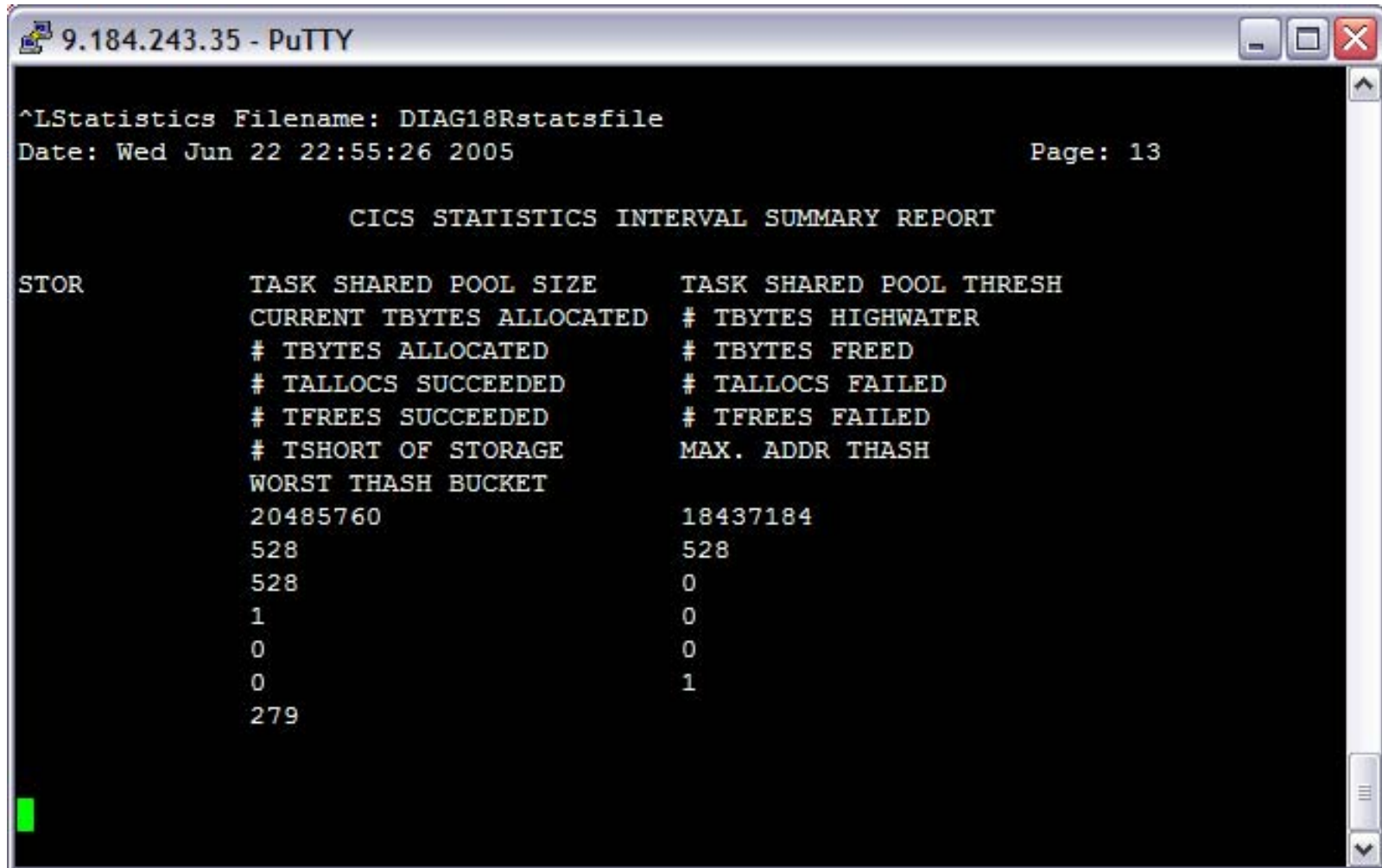
86:1:1 28x88 Command "D:\opt\var\cics_regions\IAIN\IAIN.env" [Incomplete last line] 86 lines, 4588 characters

Sample of CSMT.out



```
ERZ042006I/0116 6/21/2005 11:24:03.417513000 IAIN 5876/0001 :  
Processing terminal autoinstall request from client system '@I0K' ('@I  
0KAAAA'); terminal type is 'ibm-cics-client'  
ERZ042021I/0117 6/21/2005 11:24:03.417513000 IAIN 5876/0001 :  
Terminal 'HGYP' autoinstall is complete for client system '@I0K' ('@I0  
KAAAA'); terminal NETNAME is 'HGAAAAYP', terminal model is 'cicscli'  
ERZ016050W/0234 6/21/2005 11:24:06.602092000 IAIN 5876/0001 HGYP:  
Logical unit of work for transaction 'JHE1' has been backed out; Distr  
ibuted Transaction Service (TRAN) reason 'ENC-tra-1025: A client (not  
the transaction service) aborted'  
ERZ042028I/0159 6/21/2005 11:28:50.370131000 IAIN 5892/0001 :  
Terminal 'HGYP' with NETNAME 'HGAAAAYP' has been uninstalled.  
~  
~  
:  
4:1:1 14x70 Command "D:\opt\var\cics_regions\IAIN\data\CSMT.out" 4 lines, 774 characters
```

Sample of statsfile



9.184.243.35 - PuTTY

```
^LStatistics Filename: DIAG18Rstatsfile
Date: Wed Jun 22 22:55:26 2005 Page: 13

      CICS STATISTICS INTERVAL SUMMARY REPORT

STOR      TASK SHARED POOL SIZE      TASK SHARED POOL THRESH
          CURRENT TBYTES ALLOCATED # TBYTES HIGHWATER
          # TBYTES ALLOCATED      # TBYTES FREED
          # TALLOCs SUCCEEDED     # TALLOCs FAILED
          # TFREES SUCCEEDED     # TFREES FAILED
          # TSHORT OF STORAGE     MAX. ADDR THASH
          WORST THASH BUCKET
          20485760                18437184
          528                    528
          528                    0
          1                      0
          0                      0
          0                      1
          279
```


Traceback Files

- CICS produces traceback files when an application or CICS internal code raises an exception/signal (SIGSEGV,SIGILL).
- The traceback files are produced in the core dump directory, which is defined by the **CoreDumpName** attribute in the Region Definition (RD stanza).
 - ▶ /var/cics_regions/<region_name>/dumps/dir1 (by default)
- The generated traceback file will have a filename **cics<pid>.traceback** where :
 - ▶ pid is the process id which has generated traceback.



Contents

- Sources of Information
- Understanding Product Logs
- **CICS Product Dump and Trace**
- CICS Supplied Transactions for Debugging
- Handling Exceptions in CICS
- Sample Problem Cases and Problem Determination
- Supplying IBM Support with Information



CICS Product Dump

- Provides a detailed snapshot of what was happening in CICS at the moment the dump occurred.
- For example, you can look for the following information:
 - ▶ Number and List of tasks running in the region
 - ▶ Number and List of application servers which are idle
 - ▶ Number and List of tasks that are queued and not yet scheduled



Types of CICS Dump – Transaction Dump

- A **transaction** dump writes specified areas of memory to a file, to assist you in debugging an application program or to identify why an abnormal termination or storage violation occurred.
- Transaction dumps can be generated by,
 - ▶ EXEC CICS DUMP
 - ▶ EXEC CICS ABEND
 - ▶ Abnormal termination of transactions such as ASRA/ASRB abends
 - ▶ Not all abends will generate a dump



Types of CICS Dump – System Dump

- A **system** dump writes information about the entire CICS region, including:
 - ▶ Details of the last CICS command executed
 - ▶ Details of each transaction in progress
 - ▶ The region configuration at the time the dump is taken
 - ▶ Transaction dump of any non-application server processes
 - ▶ Information from the Encina client
 - ▶ All enabled trace information
- System dumps can be generated by,
 - ▶ CICS system shutdown
 - ▶ Abnormal CICS system terminations
 - ▶ CEMT PERFORM SNAP
 - A SNAP dump writes information about the entire CICS region and can be initiated through a CICS terminal



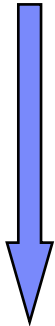
Types of CICS Dump – Core Dump

- In exceptional circumstances CICS can generate an operating system **core** dump rather than a CICS formatted dump
- Messages are written to the console giving the reason for the core dump and the file name
- CICS will attempt to format the dump using showProcInf



Dump File Name

<aaaa><nnnn>.dmp<mm>



Indicates if dump fits in one or multiple files

- Dump sequence number
- Incremented with each dump performed
- Current number saved at region shutdown for next restart

- ASRA - As a result of an ASRA abnormal termination
- ASRB - As a result of an ASRB abnormal termination
- SYSA - As a result of an SYSA abnormal termination
- SHUT - Perform a shutdown request
- SNAP - From a CEMT PERFORM SNAP DUMP request
- Four letter dumpcode from an EXEC CICS DUMP command
- Four letter abnormal termination code from an EXEC CICS ABEND command or abnormal transaction termination initiated by CICS



Setting Dump Destination

- Dump destination is controlled by the following settings
 - ▶ **DumpName** attribute of the RD
 - The directory (containing the subdirectories) to which CICS dumps are written.
 - Defaults to **/dumps**
 - ▶ **CoreDumpName** attribute of the RD
 - **Subdirectory** of the **DumpName** directory. CICS uses this subdirectory for a core dump in the event of a nonrecoverable CICS abnormal termination.
 - Defaults to **/dir1**
- By default CICS dumps will be placed in the following directory
 - ▶ **/var/cics_regions/<regionName>/dumps/dir1**



Enabling Dumps

1. By setting the following attributes in RD and TD stanza file.
 - ▶ SysDump (RD). Produce dump on shutdown, SNAP and ASRx abends.
 - ▶ ABDump (RD). Produce system or transaction dump on ASRB abends.
 - ▶ PCDump (RD). Produce system or transaction dump on ASRA abends.
 - ▶ TransDumpTrace (RD). Include system trace buffer in transaction dumps.
 - ▶ TransDump (TD). Produce transaction dump if transaction abends.
2. Using CEMT during runtime.
 - ▶ CEMT INQUIRE/SET DUMP
 - ▶ CEMT INQUIRE/SET DUMPOPTIONS
3. Through Dump Request User Exit (UE052017). The dump request user exit can be invoked at the following points:
 - ▶ When the EXEC CICS DUMP command is issued
 - ▶ When a transaction abend occurs
 - ▶ When a CEMT PERFORM SNAP is requested
 - ▶ When the region is shut down
 - ▶ When a system abend occurs



Controlling Dumps

If you want to:	Set the following
<p>Enable dump when any of the following:</p> <ul style="list-style-type: none"> ▪ CICS shuts down abnormally ▪ CEMT PERFORM SNAP ▪ ASRA / ASRB abend 	<p>SysDump = yes or CEMT SET DUMP ON</p>
<p>Produce a CICS system dump on ASRA abend</p>	<p>SysDump = yes or CEMT SET DUMP ON plus TransDump = yes plus PCDump = yes or CEMT SET DUMPOPTIONS PCDUMP</p>
<p>Produce a CICS system dump on ASRB abend</p>	<p>SysDump = yes or CEMT SET DUMP ON plus TransDump = yes plus ABDump = yes or CEMT SET DUMPOPTIONS ABDUMP</p>
<p>Produce a dump of CICS resource definitions and main storage areas related to a task</p>	<p>EXEC CICS DUMP</p>
<p>Produce a CICS Transaction Dump on ASRB abend</p>	<p>TransDump = yes ABDump = no or CEMT SET DUMPOPTIONS NOABDUMP</p>
<p>Produce a CICS Transaction Dump on ASRA abend</p>	<p>TransDump = yes PCDump = no or CEMT SET DUMPOPTIONS NOPCDUMP</p>



Formatting Dumps

- To format a CICS dump:
 1. Navigate to the location (or directory) where dump is generated
 2. Issue the following command:
 - **cicsdfmt {*dump file name*} > {*dump file output*}**



CICS Trace

- Tracing is a general way of collecting detailed information about how a process runs.
- Tracing consists of reporting information about events in a process; events include the calling of and return from functions, the exit of processes and other significant occurrences in the execution of a program.
- Trace information can be used to debug an application and to tune the operation of a product.
- Tracing can assist both application developers and system administrators, and it is often crucial when asking for product support.



CICS Trace - Types

CICS provides two types of tracing:

- **Application Trace**

- ▶ Exec Trace

- Traces all EXEC CICS commands with their options

- ▶ User Trace

- Allows application programmers to debug and investigate their programs. User trace is the trace of an individual task, that is written while a task is running. The user trace events are triggered through EXEC CICS ENTER command.

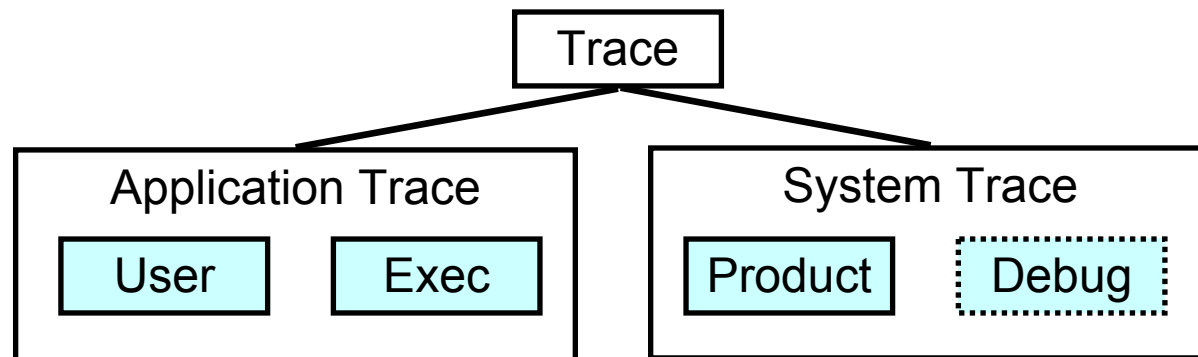
- **System Trace**

- ▶ Product Trace

- Tracks significant events in the CICS product code using 5 trace levels

- ▶ Debug trace

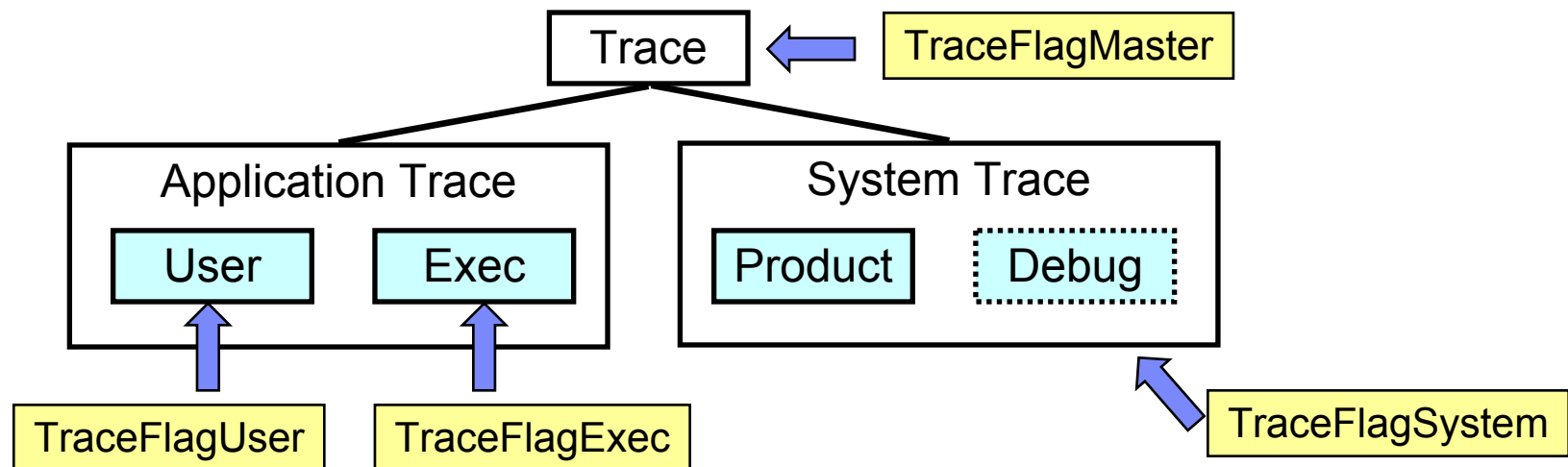
- Adds a sixth trace level. Only available through IBM support



Enabling CICS Trace

Trace specific attributes defined in RD stanza:

- **TraceFlagMaster**
Enables region-wide trace collection. If **off**, no trace is collected, regardless of other trace settings.
- **TraceFlagExec**
Enables collection of **exec** trace, a subset of application trace. If **off**, no exec trace is collected.
- **TraceFlagUser**
Enables collection of **user** trace, a subset of application trace. If **off**, no user trace is collected.
- **TraceFlagSystem**
Enables collection of **system** trace. If **off**, no system trace is collected



Enabling CICS Trace

1. Set the correct RD attributes
 - ▶ For application trace
 - **TraceFlagMaster** and
 - **TraceFlagUser** or
 - **TraceFlagExec**
 - ▶ For system trace
 - **TraceFlagMaster** and **TraceFlagSystem**
 - set the **TraceSystemSpec** RD attribute to indicate the events, modules, and processes you want to trace.
2. Set environment variable CICSTRACE
3. To collect the trace at run time
 - ▶ Call EXEC CICS TRACE ON in the application
 - ▶ Use CEMT SET TRACE ON from a terminal



Application Trace - Collection

1. Set **TraceFlagMaster** in RD stanza or use **EXEC CICS TRACE ON** command or use **CEMT SET TRACE ON**
2. Set appropriate application trace flags:
 - ▶ For user trace, set **TraceFlagUser** in RD stanza or use **EXEC CICS TRACE ON USER** command.
 - ▶ For exec trace, set **TraceFlagExec** in RD stanza or use **EXEC CICS TRACE ON EI** command.
3. Use **EXEC CICS TRACE ON USER SINGLE** command within program
 - ▶ This turns on writing of user and exec trace
 - ▶ This request is cancelled when program exits
 - ▶ Trace events triggered by **EXEC CICS ENTER** command



Application Trace - File Name and Location

- When application trace is collected, the trace records are written to files.
- Trace output directory determined by **TraceDirectoryUser** in RD stanza. This defaults to
 - ▶ /tmp on UNIX systems
 - ▶ <installDrive>:\var\cics_tmp on Windows systems
 - ▶ Output files named according to the following pattern:
 - *regionName.userName.nprocessNumber.pprocessID.cicsusr*
 - CICSREG.CICSUSER.n101.p204576.cicsusr
- Trace output directory for programs that use user trace but run without a user identifier determined by **TraceUserPublicFile** in RD stanza. For example, programs that run at startup. This defaults to **cicspubl**.
 - ▶ Output files named according to following pattern:
 - *regionName.public.cicsusr*
 - CICSREG.public.cicsusr



System Trace - Collection

1. Set **TraceFlagMaster** and **TraceFlagSystem** in RD stanza
2. Set **TraceSystemSpec** in RD stanza to indicate the events, modules, and processes to trace



System Trace – Trace Specification

```
TraceSystemSpec="TraceDefn[,TraceDefn]"
```

There are two formats for **TraceDefn**:

- *moduleList=traceLevel*
 - ▶ **moduleList** is a list of modules separated by a "+", or "all"
 - ▶ **traceLevel** is either 0 (none) or 4 (all). Levels 1, 2 and 3 not used.
- *proc=processTypeList*
 - ▶ **processTypeList** is a list of CICS process names or "all"

```
TraceSystemSpec="all=0,damtd+damfi=4,proc=as,taslu=4,proc=ip+rl,suppr+comsu=4"
```

- Turns all module trace off
- Modules: **damtd**, **damfi** in all processes
- Module: **taslu** in **as** process
- Module: **suppr**, **comsu** in **ip** and **rl** processes



System Trace – Destination, Name and Location

- Trace records sent to one of three destinations set in RD stanza:
 - ▶ **TraceFlagBuffer**. Sends trace records to the trace ring buffer; default is **on**
 - ▶ **TraceFlagAux**. Sends trace records to auxiliary files; default is **off**
 - ▶ **TraceFlagExternal** (AIX only). Sends trace records to the external trace facility; default is **off**
- Trace output directory for AUX and ring buffer dump determined by **TraceDirectorySystem** in RD stanza. This defaults to
 - ▶ */var/cics_regions/<regionName>/dumps/dir1*

Ring Buffer

- Size determined by **TraceMaxSizeBuffer** in RD stanza: default is **131072** bytes.
- Buffer dumped automatically according to environment variables (next page)
- Output files named according to the following pattern:
 - *regionName.processName.processNumber.cicsdmp.indexCounter*

AUX Trace

- Sent directly to file. Size determined by **TraceMaxSizeAux** in RD stanza; default is **0** (unlimited)
- Output files named according to the following pattern:
 - ▶ *regionName.processName.nprocessNumber.pprocessID.tthreadID.cicstrc*



System Trace – Dumping Ring Buffer Trace

- **CICSTRACE_DUMP_ON_ABNORMAL_EXIT**
Dumps ring buffer whenever a task exits through an abnormal task-termination procedure. Set the variable to any value.
- **CICSTRACE_DUMP_ON_EXIT**
Dumps ring buffer whenever a task exits through the normal or abnormal task-termination procedure. Set the variable to any value.
- **CICSTRACE_DUMP_ON_SYMREC**
Dumps ring buffer whenever a symrec is recorded. Set the variable to any value; the value itself is not meaningful.
- **CICSTRACE_DUMP_ON_ABEND**
Dumps ring buffer whenever a symrec is generated for one of the CICS abend codes. Set the variable to a comma-delimited list of abend codes.
- **CICSTRACE_DUMP_ON_MSN**
Dumps ring buffer whenever a symrec is generated for one of the CICS message sequence numbers (MSNs). Set the variable to a comma-delimited list of MSNs.



CICSTRACE Environment Variable

CICSTRACE=

"-d TraceDirectorySystem	-u TraceDirectoryUser
-A TraceFlagAux	-B TraceFlagBuffer
-E TraceFlagExec	-X TraceFlagExternal
-M TraceFlagMaster	-a TraceMaxSizeAux
-b TraceMaxSizeBuffer	-t TraceSystemSpec
-p TraceUserPublicFile"	

CICSTRACE=" -A on -u /tmp -B off -M on -S on -t all=4"

- Set in region **environment** file to override trace-related data in RD stanza
- Set on command line to trace standalone processes and commands
- Setting "CICSEXTERNALTRACE=1" has same effect as CICSTRACE for standalone processes



System Trace – Dynamic Activation

- To start collecting system trace from a running region
 1. CECI TRACE ON
 2. CECI TRACE SYSTEM ON
 3. CEMT SET AUXTRACE ON

- To stop collecting system trace from a running region
 1. CEMT SET AUXTRACE OFF
 2. CECI TRACE SYSTEM OFF
 3. CECI TRACE OFF



Summary of trace-related RD stanza entries

- Region-wide trace: **TraceFlagMaster** (off)
- Application trace
 - ▶ Trace attributes
 - **TraceFlagUser** (off)
 - **TraceFlagExec** (off)
 - ▶ Trace storage
 - **TraceDirectoryUser** (/tmp or \var\cics_tmp)
 - **TraceUserPublicFile** (cicspubl)
- System trace: **TraceFlagSystem** (off)
 - ▶ Trace specification: **TraceSystemSpec** (all=0)
 - ▶ Trace destinations
 - **TraceDirectorySystem** (/var/cics_regions/regionName/dumps/dir1)
 - Buffer
 - **TraceFlagBuffer** (on)
 - **TraceMaxSizeBuffer** (131,072)
 - Auxiliary files
 - **TraceFlagAux** (off)
 - **TraceMaxSizeAux** (0, which means unlimited size)
 - External: **TraceFlagExternal** (off)



Formatting CICS Trace

- To format a CICS trace:
 1. Navigate to the location (or directory) where trace is generated
 2. Issue the following command:
 - **`cicstfmt {cicstrace file name} > {cicstrace file output}`**



Contents

- Sources of Information
- Understanding Product Logs
- CICS Product Dump and Trace
- **CICS Supplied Transactions for Debugging**
- Handling Exceptions in CICS
- Sample Problem Cases and Problem Determination
- Supplying IBM Support with Information



CICS Supplied Transactions for Debugging

CICS supplies the following transactions to aid application program debugging

- **CADB**

To configure and debug Micro Focus and Acucobol-GT COBOL programs
[Unix only]

- **CDCN**

To configure and debug C/C++/IBM COBOL/Acucobol-GT based
application programs. [AIX only]

- **CJDB**

To configure and debug Java application programs

- **CEDF**

Execution diagnostic facility. You can use CEDF interactively to follow the
internal flow from one CICS command to another.



CADB – Some Details

- CADB changed and enhanced with TXSeries 5.1
 - ▶ Modelled on CDCN
 - ▶ Different compile options used
 - ▶ Configuration stored in memory rather than file
 - ▶ Configuration isolated on a per user and per key basis (was per user only)
- The **AllowDebugging** flag in the RD stanza must be set to **'yes'** before executing CADB.
- Debugging configuration can be turned on and turned off using the CADB.
- CADB and Animator both need same key
 - ▶ Allows individual users to control Animator process, even if several users share a CICS user.
- CADB itself does not start Animator. Animator is started only when a COBOL program is run later.
- CADB is also used for Acucobol-GT



CADB – How to Use with Animator

1. Set **AllowDebugging=yes** in the RD stanza.
2. Make sure Cobol program has been built for debug (**cicstcl -a**).
3. Enter **CADB [key]** transaction
 - ▶ Key, is an optional unique 8 character string used to distinguish between different debug configurations under the same CICS user.
 - ▶ Means two people both signed on as CICSUSER can debug separately
4. Enter a value for **COBANIMSRV Id**. A character string (max of 40), required to associate a COBOL program with an Animator process running under a different Window.
5. Optionally enter a **terminal, sysid, transaction** or **program** to control which program to debug. Terminal is highest order of precedence.
6. **Enter** to comit selection, then **PF3** to exit.
7. Open another window (X-Window or telnet). Setup the basic Cobol environment variables like COBDIR etc.
8. Set an environment variable called **COBANIMSRV**, with the value of the field **COBANIMSRV Id** in the CADB Screen.
9. Run **cicsanimsrv** in the same window
 - ▶ This is a CICS supplied program written as a wrapper to **cobanimsrv** to start it with the correct permissions.
10. Invoke the debug program. If the correct conditions are met Animator will intercept



CADB – How to Use with Acucobol

1. Set **AllowDebugging=yes** in the RD stanza.
2. Make sure Cobol program has been built for debug (**cicstcl -a**).
3. Enter **CADB ACU** transaction
4. Enter the following values
 - DEBUG=Yes
 - DISPLAY=<IP address of the Xserver host>:DisplayNumber
5. Optionally enter a **terminal**, **sysid**, **transaction** or **program** to control which program to debug. Terminal is highest order of precedence.
6. **Enter** to comit selection, then **PF3** to exit.
7. Invoke the debug program. If the correct conditions are met Acucobol-GT Debugger will start in a separate window.



Debugging on Windows

- Compile options and steps vary with language and vendor.
- Refer to Application Programming Guide, Chapter 10 for details.



Contents

- Sources of Information
- Understanding Product Logs
- CICS Product Dump and Trace
- CICS Supplied Transactions for Debugging
- **Handling Exceptions in CICS**
- Sample Problem Cases and Problem Determination
- Supplying IBM Support with Information



Handling Exceptions in CICS

- In a CICS environment exceptions could be raised in any of the following modules:
 - ▶ User application program
 - ▶ Other IBM or Third-party product modules loaded in the CICS application server process
 - ▶ CICS product module

- Exceptions could generate the following docs, which provides details on where the exception occurred:
 - ▶ /var/cics_regions/<region_name>/dumps/dir1/cics<pid>.traceback
 - ▶ /var/cics_regions/<region_name>/dumps/dir1/core.<timestamp>



Handling Exceptions in CICS

- User application program
 - ▶ CICS reports ASRA/ASRB abend
 - ▶ User application can write ABEND HANDLER to do any clean up functions during exceptions like closing files, freeing allocated resources etc,.
- Other IBM or Third-party product modules loaded in the CICS application server process
- CICS product module
 - ▶ Exceptions generated within CICS product module are handled by CICS and would require IBM support to further investigate the cause for the exception.



Contents

- Sources of Information
- Understanding Product Logs
- CICS Product Dump and Trace
- CICS Supplied Transactions for Debugging
- Handling Exceptions in CICS
- **Sample Problem Cases and Problem Determination**
- Supplying IBM Support with Information



Problem 1: ASRA/ASRB Abend

- ASRA/ASRB abend is caused due to exceptions raised within the USER CODE. The exceptions could be SIGSEGV, SIGBUS, SIGILL etc,. The exceptions could be raised in other involved products.

```
int main ( void )
{
    char *ptr; /* Un-Initialized pointer */

    strcpy (ptr, "Crash me!");

    EXEC CICS RETURN;
}
```



Problem 1 : Problem Determination

- The console.nnnnnn log would indicate if a traceback file is generated subsequent to the ASRA/ASRB messages.
- The traceback file would indicate the failing transaction / program and the function stack along with the offset where the exception is raised. It would also indicate as to what signal/exception is raised within the failing program.
- Review the application program as mentioned in the traceback file at the specified offset and determine the cause for the problem.



Problem 2: CICS Region Stalled

- CICS can stall during initialization, when it is running apparently normally, or during termination.
 - ▶ No new client connections can be established
 - ▶ The region is short on storage storage, so existing tasks are waiting for storage to become available.
 - ▶ The system is at the MaxServer limit and no new tasks may start.
 - ▶ Transaction waiting for resources indefinitely
 - ▶ Deadlock on resources



Problem 2 : Problem Determination

- Ensure that the region is not stalled due to user transaction occupying all the application servers. This can be verified through CEMT I TASK.
- Ensure you have set proper timeouts for the transactions or communication links to avoid 'run-away' situations.
- Ensure you have no applications waiting for resources indefinitely.
- If no user transactions are not involved in the region stall issue, then it could be an internal product issue. Contact IBM Support representative for further problem diagnosis.



Problem 3: A147 Abend

- Following are the possible reasons why A147 abend is reported by CICS server:
 - ▶ PURGE/FORCEPURGE of a CICS task
 - ▶ ECI client timeout
 - ▶ Transaction timeout
 - ▶ Application server unexpectedly ended
 - ▶ Network failures



Problem 3 : Problem Determination

- A147 is an expected abend code if user issues a CEMT SET PURGE / FORCEPURGE command against a running task.
- A147 is an expected abend code if the client or transaction times out.
- A147 abend is reported in cases where application server processes dies unexpectedly. This would generally leave a core (i.e. the process image) in the regions dumps directory which can be further investigated to determine the cause.
- A147 abend could occur due to network failures, If so corresponding messages in the console.nnnnnn log would indicate the communication error code.



Problem 4: Diagnosing Memory Leaks

- Memory leaks are often caused due to lack of freeing up of resources either by the application or by CICS. Following are the memory pools where applications could allocate resources from:
 - ▶ Task Shared Pool
 - ▶ Task Private Pool / Processes Heap
 - ▶ Process File Descriptors
 - ▶ Region Pool



Problem 4 : Problem Determination

- The CICS_LEAKDEBUG environment variable provides a facility that enables you to generate reports about memory and file descriptor leaks that can occur in CICS Application Server processes.
- You can use this facility when:
 - ▶ Continuous growth in CICS Application Server process memory is observed over a period of time. On an Open Systems platform, use the ps command to monitor process memory size; on a Windows platform, use Task Manager.
 - ▶ Recurring messages, such as Task Shared Pool is under stress. CICS is short on Task Shared storage, are seen in the Console log. These messages indicate growth in the Task Shared Pool.
 - ▶ A Memory leak occurs in the Task Private pool. Note: Failure of GETMAIN API with a NOSTG condition indicates that CICS has no more Task Private pool storage available.
 - ▶ Too many file descriptors remain open in the system.



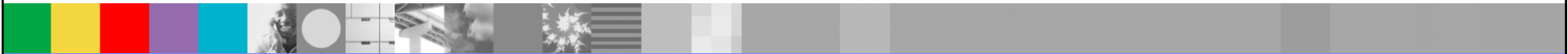
Problem 4 : Problem Determination

- You can use CICS_LEAKDEBUG facility to isolate leaks in your application programs. The debugging reports contain information that is recorded at the following places in CICS code:
 - ▶ User application load and unload
 - ▶ User application entry and exit
 - ▶ EXEC CICS API entry and exit
 - ▶ A report is generated in a set of text files for every CICSAS process and stored in the specified path. Every file is named in the format, cicsas.<process id>



Contents

- Sources of Information
- Understanding Product Logs
- CICS Product Dump and Trace
- CICS Supplied Transactions for Debugging
- Handling Exceptions in CICS
- Sample Problem Cases and Problem Determination
- **Supplying IBM Support with Information**



What IBM Needs

- Information about the problem:
 - ▶ Is it a new problem or a further call on an existing problem?
 - Brief description of the problem
 - Severity of the problem:
 - You cannot use CICS, which results in a critical condition that needs immediate attention.
 - You can use CICS, but that operation is severely restricted.
 - You can use CICS, but with limited functions. However, the problem is not critical to your overall operation.

- About your environment:
 - ▶ Version and release number of:
 - CICS
 - Operating Systems
 - Other IBM or third-party software products working in the problem environment



What IBM Needs

CICS Region information:

- console.<nnnnnn> file
- symrecs.<nnnnnn> file
- console.msg file
- console.nam and symrecs.nam files
- <regionname>.env file
- cics<pid>.traceback files, if any
- core<pid>.timestamp or core.<timestamp>.fmt files, if any
- Permanent database stanza definition files
- A copy of the /<region>/environment file
- The contents of the /data directory
- Any dump files found in the /dumps directory



What IBM Needs

Following are some of the problem specific document collections:

- Problem : CICS Stalls or Hangs
- Snapshot of all the CICS processes running in the region. Snapshots can be taken with the help of showProclInfo tool supplied with the product.
- Problem: Resource or Memory Leaks
- Enable and generate debug reports through CICS_LEAKDEBUG tool.
- Problem: Abnormal region shutdown ex. U5701, U5655 abends
- Enable system dump and provide formatted system dump outputs.
- Problem: XA Resource manager errors encountered causing CICS region down
- Enable third-party resource manager debug logs. For example to specify debug flags in the XA_OPEN string.



Use CICSERVICE Tool

- Provides all basic information that IBM needs.
- To use:
 - ▶ Format dumps using cicsdfmt. Place in /dumps directory.
 - ▶ Format any core<pid>.timestamp core files using showProcInfo. Place in /dumps directory.
 - ▶ Format any trace files using cicstfmt and place in /dumps directory
 - ▶ Run command `/$CICS_HOME/utis/cicsservice`
 - ▶ Select everything (option 3)
- Output placed in:
 - ▶ `/tmp/cservice.tar.Z` (Open systems)
 - ▶ `%tmp%/cservice.tar` (Windows)



Pointers on Avoiding Problems

- Good software engineering practice
- Share problem repositories
- Maintain common knowledge base database
- Have change management and controlled library system
- Maintain common software stack
- Follow programming standard
- Keep track of changes that are being made to the production system



Reference

- CICS Problem Determination Guide
 - ▶ This book is intended to help you determine the causes of problems in a CICS® system. The book provides information to help you resolve CICS application and system problems, and tells you how to use your support organization.
- Support Web page
 - ▶ <http://ibm.com/software/http/txseries/support>
 - PTFs, Hints & Tips, Technotes, Fixes, Patches, Supported Software Stack

