Debugging Aids - ADL Trace Facility

The Adabas Bridge for DL/I includes a Trace facility with which you can gather information on the following:

- DL/I and Adabas calls (referred to as "database call trace").

- Internal routine ADL calls (referred to as "routine trace")

You may activate the Trace facility in different ways:

- By specifying the `TRACE` parameter in the `ADL` parameter module (see the section *ADL Parameter Module* in the *ADL Installation* documentation).

- By specifying the TRACE parameter as one of the parameters for the ADL initialization program, DAZIFP (see the section *Batch Installation and Operation* in this documentation for details), or

- Dynamically under CICS in the CICS Maintenance of the ADL Online Services or by running the DAZT transaction.

This chapter covers the following topics:

- Data Base Call Trace

- Internal Routine Trace

- z/OS JCL Requirements

- z/VSE JCS Requirements

Data Base Call Trace

Tracing DL/I and Adabas Calls

The Trace facility for DL/I and Adabas calls consists of two parts, one that is responsible for data acquisition and the other for reporting.

Data Acquisition

The data acquisition part of the Trace facility collects data for every DL/I and Adabas call while the application program is running, and stores them in a sequential file known as the trace file.

The default setting causes data to be collected for all calls, from the beginning of the job to the end. However, you may also use the TRACE parameter operands listed below to trace a specific number of DL/I calls or to start a trace at a particular call.

When you activate the Trace facility under CICS with the ADL Online Services, the system prompts you for a terminal ID and the type of trace wanted. Specifying a terminal ID results in a trace of calls from the specified terminal only. Any blank character in the specified terminal ID is treated as a wildcard, i.e. it matches all characters. Thus, if a blank terminal ID is specified (4 blank characters) a trace of calls from all active terminals is conducted.

The syntax of the Trace parameter for DAZIFP or the ADL Parameter Module is as follows:

```
TRACE=([t],[s],[a],[n],[m],[c],[u],[b])
```

All parameters are positional parameters. Trailing commas can be omitted, e.g.

```
TRACE=(RD)
```

The operands of the TRACE parameter have the following meaning:

Operand	Explanation
t	The type of trace requested. One or more of the following values may be specified: **R**: All internal routine calls will be traced. **D**: All DL/I and Adabas calls will be traced. **U**: Special trace for the CBC utility. The order in which these values are specified is irrelevant. Default: None
s	The start, i.e. the first DL/I call to be traced. Possible values: 1 - 9999999 Default: 1
a	The number of DL/I calls to be traced. Possible values: 1 - 9999999 Default: 9999999
n	The name of the routine at which the routine trace is to be started. The name given must start with "DAZ", and may have a maximum of 8 characters. Default: DAZCOMDC
m	The number of the call to the routine specified in "n" at which the routine trace is to be started. An entry of 0 causes "n" to be ignored and the complete routine trace to be printed. Default: 0
c	The name of the routine which is to be counted. The name given must start with "DAZ", and may have a maximum of 8 characters. Default: DAZCOMDC
u	The logical unit number (for z/VSE only). Possible values: 0-999 Default: 11
b	The block size (for z/VSE only). Possible values: 0-99999 Default: 8196

For the ADL CALLDLI Interface the data collected contain the following information:

● The "before" image of a DL/I call (i.e. its form before it is passed to the system): the function, SSAs and I/O area (for insert and replace calls) are traced.

● The "after" image of an Adabas call (i.e. its form when it is returned by the system): the Adabas control block and buffers (any of the format, record, search, value and ISN buffers, if used) are traced. The fields of the Adabas control block are listed individually in readable format (character or numeric) and in hexadecimal format (denoted by a leading "x"). The buffers are listed in hexadecimal

and in character format.

- The "after" image of a DL/I call: the user PCB, key feedback area and I/O area (for retrieval calls) are traced.

```
000001 GU
          SSA1  :C3D6E4D9 E2C54040 4DC3D6E4 D9E2C5D5 D67E40D4 C1E3C8C5 D4C1F2F0 F05D          *COURSE  (COURSENO= MATHEMA200) *
----------------------------------------------------------------------------------------------------------------------
L6   ISN:  0000000021 x00000015   DBID: 01424 x0590   FNR:  00833 x0341     RSP: 00000 x0000  CID: 'Z???' xE9010101 TERM:
     TIME: 0000000005 x00000005   ISNL: 0000000000    ISNQ: 0000000000      OP1: 'R'   xD9  OP2: 'V'    xE5      TASK:
     ADD1: 'AA' C1C1BBCA40404040  ADD2: ' ?' 001E003F ADD3: 4040404040404040 ADD4: 4040404040740590 ADD5: 0000000000000000
     FBL: 0024 FB: F4E76BF4 E76BE9F4 6BF8E76B F8E76BF4 E76BF4E7 6BE2C14B                     *4X,4X,Z4,8X,8X,4X,4X,SA.       *
     RBL: 0063 RB: 40404040 40404040 00404040 40404040 40404040 40404040 40404040 40404040  *        .                       *
                   40D4C1E3 C8C5D4C1 F2F0F0D4 C1E3C8C5 D4C1E3C9 C3E24040 40404040 404040     * MATHEMA200MATHEMATICS          *
     SBL: 0003 SB: C1C14B                                                                    *AA.                            *
     VBL: 0010 VB: D4C1E3C8 C5D4C1F2 F0F0                                                    *MATHEMA200                     *
----------------------------------------------------------------------------------------------------------------------
     DBD: COURSEDB STA:     SEG: COURSE   LEV: 01 SEN: 00000004 OPT: AP   RES: 01030470
     KYL: 0010 KY: D4C1E3C8 C5D4C1F2 F0F0                                                    *MATHEMA200                     *
     IOL: 0030 IO: D4C1E3C8 C5D4C1F2 F0F0D4C1 E3C8C5D4 C1E3C9C3 E2404040 40404040 4040       *MATHEMA200MATHEMATICS          *
```

Database call trace

For the ADL Consistency Interface, the following data are collected:

- A "before" image of the user Adabas call.

- A "before" image of internally emulated DL/I type calls. Only the I/O area but no SSAs will be traced.

- An "after" image of Adabas calls issued by ADL i.e. in its form when it is returned from Adabas.

- An "after" image of internally emulated DL/I type calls.

- An "after" image of the user Adabas call. This is after the Consistency has completed its processing, but before the call is either passed to Adabas or control is returned to the user. Note that a non-zero Adabas return code displayed is the actual ADL return code (in a range from 1201 to 1299) and not the return code which will be seen by the user (216).

Only those user Adabas calls will be traced which perform an update function (insert, delete or replace) and reference to an ADL file.

For the ADL Consistency Interface it is recommended to overwrite the default routine name of the 'n' and 'c' operands with the ADL routine name DAZCFDEC.

Data Reporting

The data reporting part of the Trace facility reads and interprets the data stored in the trace file and produces a trace report. As such, it is part of the ADL nucleus and is activated by executing the ADL initialization program DAZIFP with the parameters shown below as provided by the file DAZIN1 (see the section *Batch Installation and Operation* for details).

```
PRT,DAZPRINT
```

You can control the Trace data reported using an additional control statement read in from either DAZIN1 (z/OS) or DAZIN2 (z/VSE). This statement has one of the following two formats and options:

```
MODE=SHORT[,TERM=t][,TASK=i]
```

or

```
MODE=FULL[,ADA=x][,DLI=y][,TERM=t][,TASK=i]
```

where

Operand	Explanation
t	is a four-character CICS terminal ID indicating that only DL/Iand Adabas calls originating from the specified terminal will be repeated.
i	is a task number up to five digits long indicating that only DL/Iand Adabas calls originating from the specified task number will be repeated.

x is any combination of the following values:

Value	Suppressed Buffer/Calls
R	Adabas record buffer
F	Adabas format buffer
S	Adabas search buffer
V	Adabas value buffer
I	Adabas ISN buffer
N	Adabas calls before the first DL/I call

and *y* is any combination of the following values:

Value	Suppressed Areas
I	DL/I I/O area
K	DL/I key feedback area
P	DL/I user PCB
S	DL/I SSAs

Specifying any of the x or y parameter values suppresses reporting of the indicated buffer or area.

Note:
MODE=SHORT

is the same as

```
MODE=FULL,ADA=RFSVI,DLI=IKS
```

Example:

```
MODE=FULL,ADA=RFI,DLI=I
```

This parameter statement would cause the Adabas search and value buffers and the DL/I PCB, key feedback area and SSAs to be reported.

Internal Routine Trace

The Trace facility may be used to trace all internal routine calls for ADL. The default setting causes all routine calls from the beginning of the job to the end to be traced. However, you may use the TRACE parameter operands listed earlier in the section on Data Acquisition to specify the name of the routine at which tracing should be started and to ignore the first "n" times the routine was called.

In batch, the routine calls trace is routed direct to the printer. In CICS, an extra partition data set is used for this. Any standard printing utility, or the DAZPRINT utility, may then be used to print the trace. The DAZPRINT utility is activated by executing the ADL initialization program DAZIFP with the parameters shown below (see the section *Batch Installation and Operation* for details):

```
PRT,DAZPRINT
```

The following additional parameter statement must be specified:

```
MODE=ROUTINE[,TERM=t]
```

where

t is a four-character CICS terminal ID, indicating that only routine calls originating from the terminal specified should be printed.

z/OS JCL Requirements

The JCL requirements for the Trace facility are discussed below. Explanations of how to use the Trace facility during a batch run and how to print out the trace data set are given.

Using the Trace Facility

The table below lists the data sets used by the ADL batch monitor when the Trace facility has been activated. These data sets are used in addition to any others.

DDname	Medium	Description
DAZOUT5	Tape/Disk	Trace output file *
DAZOUT1	Printer	ADL routine calls trace **

* Only required when the trace to be performed is to include DL/I and Adabas calls.

** Required when the trace to be performed is to include internal ADL routine calls. This file is also used to print error messages, and should therefore already have been included.

Example

```
//        EXEC PGM=DAZIFP,PARM='DLI,pgmname,psbname,TRACE=(RD)'
//STEPLIB  DD DSN=ADL.LOAD,DISP=SHR
//        DD DSN=ADABAS.LOAD,DISP=SHR
//DDCARD   DD *
ADARUN PROGRAM=USER,...
//DAZOUT1  DD SYSOUT=X
//DAZOUT5  DD DISP=SHR,DSN=d-trace-dataset
```

Printing the Trace Data Set

The following table lists the data sets used by the Print Trace utility, DAZPRINT.

DDname	Medium	Description
DAZIN1	Reader	Control input for the Print utility, DAZPRINT.
DAZOUT1	Printer	Report, messages and codes.
DAZIN4	Tape/Disk	DL/I and Adabas calls trace file.
DAZIN5	Tape/Disk	Routine trace file.

Examples

The following is an example of a job to print the data base call trace:

```
//        EXEC PGM=DAZIFP,PARM='PRT,DAZPRINT'
//STEPLIB  DD DSN=ADL.LOAD,DISP=SHR
//DAZIN4   DD DSN=d-trace-dataset,DISP=SHR
//DAZOUT1  DD SYSOUT=X
//DAZIN1   DD *
MODE=FULL
//*
```

The following is an example of a job to print the internal routine trace from disk:

```
//        EXEC PGM=DAZIFP,PARM='PRT,DAZPRINT'
//STEPLIB  DD DSN=ADL.LOAD,DISP=SHR
//DAZIN5   DD DSN=r-trace-dataset,DISP=SHR
//DAZOUT1  DD SYSOUT=X
//DAZIN1   DD *
MODE=ROUTINE
//*
```

z/VSE JCS Requirements

The JCS requirements for the Trace facility described in this section are discussed on the following pages. Explanations of how to use the Trace facility during a batch run and how to print out the trace data set are given.

Using the Trace Facility

The table below lists the files used by the ADL batch monitor when the Trace facility has been activated. These files are used in addition to any others.

DTF	Logical Unit	Medium	Description
DAZOT5D	SYS012	Disk	Trace output file. *
DAZOT5T	SYS012	Tape	Trace output file. *
DAZOUT1	SYSLST	Printer	ADL routine calls trace. **

* Only required when the trace to be performed is to include DL/I and Adabas calls. Only one of the two files is needed.

The logical unit indicated is the default logical unit. To change it, specify the TRACE parameter either in the ADL parameter module or as a dynamic parameter:

TRACE=(,,,,,,6)

** Required when the trace to be performed is to include internal ADL routine calls. This file is also used for the printing of error messages, and is therefore automatically included.

Example

```
// ASSGN SYS012,DISK,VOL=volume,SHR
// DLBL DAZOT5D,'d-trace-dataset'
// EXTENT SYS012,volume,.....
// EXEC PROC=ADLLIBS
// EXEC DAZIFP
DLI,pgmname,psbname,TRACE=(RD)
/*
ADARUN PROGRAM=USER,.....
/*
/&
```

Printing the Trace Data File

The following table lists the files used by the Print Trace utility, DAZPRINT.

DTF	Logical Unit	Medium	Description
DAZIN1	SYSIPT	Reader	Control input for the ADL batch monitor, DAZIFP.
DAZIN2	SYSIPT	Reader	Control input for the Print utility, DAZPRINT.
DAZIN4D	SYS012	Disk	DL/I and Adabas call trace file. *
DAZIN4T	SYS012	Tape	DL/I and Adabas call trace file.*
DAZOUT1	SYSLST	Printer	Report, messages and codes.
DAZIN5D	SYS014	Disk	Routine trace file. **
DAZIN5T	SYS014	Tape	Routine trace file. **

* Only one of the two files is required. The logical unit indicated is the default logical unit. To change it, specify the TRACE parameter either in the ADL parameter module or as a dynamic parameter:

TRACE=(,,,,,,6)

** Only one of the two files is required. The logical unit indicated is the default logical unit. To change it, specify the FX parameter in the ADL parameter module:

FX=(15,)

The control input for both the batch monitor (DAZIFP) and DAZPRINT is read from SYSIPT. The control statements must be specified in the following order:

```
PRT,DAZPRINT                          input for DAZIFP
/*
MODE=FULL                             input for DAZPRINT
/*
```

Examples

The following is an example of a job to print the data base call trace:

```
// ASSGN SYS012,DISK,VOL=volume,SHR
// DLBL DAZIN4D,'d-trace-dataset'
// EXTENT SYS012,volume
// EXEC PROC=ADLLIBS
// EXEC DAZIFP
PRT,DAZPRINT
/*
MODE=FULL
/*
/&
```

The following is an example of a job to print the internal routine trace:

```
// ASSGN SYS014,DISK,VOL=volume,SHR
// DLBL DAZIN5D,'r-trace-dataset'
// EXTENT SYS014,volume
// EXEC PROC=ADLLIBS
// EXEC DAZIFP
PRT,DAZPRINT
/*
MODE=ROUTINE
/*
/&
```