DBLOG Trace Screen

The **DBLOG Trace** screen displays recorded log data on Adabas commands, or DL/I and SYNC/ROLB calls, or SQL statements which are kept in the Natural DBLOG buffer.

This section covers the following topics:

- DBLOG Trace Screen for Adabas Commands
- DBLOG Trace Screen for DL/I Calls
- DBLOG Trace Screen for SQL Statements

DBLOG Trace Screen for Adabas Commands

- Invoking DBLOG Trace for Adabas Commands
- Screen Columns and Commands on DBLOG Trace
- Displaying Adabas Buffers
- Displaying Adabas Commands that use Multi-Fetch

Invoking DBLOG Trace for Adabas Commands

The following is an example instruction for invoking the **DBLOG Trace** screen for Adabas commands.

1. Write the following Natural program:

```
DEFINE DATA LOCAL
1 EMP-VIEW VIEW OF EMPLOYEES
  2 NAME
END-DEFINE
READ (3) EMP-VIEW BY NAME
  DISPLAY NAME
END-READ
END
```

2. Enter the following Natural system command

```
TEST DBLOG
```

   The message `DBLOG started now` is displayed.

3. Enter the following Natural system command:

```
RUN
```

The Natural program in the source area is executed.

4. Enter again:

```
TEST DBLOG
```

Logging is deactivated and a **DBLOG Trace** screen similar to the example below appears:

```
14:14:23             ***** NATURAL TEST UTILITIES *****             2008-07-31
User SAG                        - DBLOG Trace -                  Library SAG
M   No Cmd   DB   FNR  Rsp        ISN        ISQ  CID CID(Hex) OP Pgm      Line
_    1 L3    10   316            5555             &?? 00500101  A LOGTEST  0050
_    2 L3    10   316            5557             &?? 00500101  A LOGTEST  0050
_    3 L3    10   316            2108             &?? 00500101  A LOGTEST  0050
_    4 RC    10   316                             &?? 00500101 SI LOGTEST  0050
_    5 RC    10                                      00000000 F  LOGTEST  0080


Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Print Exit        Posi  --    -     +     ++                Canc
```

Screen Columns and Commands on DBLOG Trace

This section describes the columns of fields contained in the **DBLOG Trace** screen and the commands available to scroll in the screen or in a buffer window opened from the screen (see *Displaying Adabas Buffers*). You execute a command by either pressing a PF key or entering a direct command in the Command line.

Column	PF Key Direct Command	Explanation
M		Input option for line commands that invoke extra windows with detailed information on buffers: see *Displaying Adabas Buffers*.
No		Sequence number. The commands are displayed in the sequence in which they were executed.
Cmd		Adabas command.
DB		Database ID.
FNR		File number.
Rsp		Adabas response code.
ISN		Internal sequence number of record.
ISQ		ISN quantity.
CID		Command ID.
CID (Hex)		Command ID in hexadecimal format.
OP		Adabas Command Options 1 and 2.

Column	PF Key Direct Command	Explanation
Pgm		Program name.
Line		Source code line number.
	PF2	Prints a hardcopy of a screen shot.
	PF3	Exits the **DBLOG Trace** screen or closes a buffer window. The current log records are kept in the Natural DBLOG buffer.
	PF5	Moves log entries to the top of the screen: In column **M**, position the cursor next to the desired command and sequence number listed in column **No** and choose PF5. The logs are repositioned starting with the sequence number selected.
	PF6 or - -	Scrolls to the beginning of a list or the data in a buffer window.
	PF7 or -	Scrolls up one page in a list or the data in a buffer window.
	PF8 or +	Scrolls down one page in a list or the data in a buffer window.
	PF9 or ++	Scrolls to the end of a list or the data in a buffer window.
	PF10	Only available in a buffer window with multiple record/format buffers. Displays the previous record/format buffer.
	PF11	Only available in a buffer window with multiple record/format buffers. Displays the next record/format buffer.
	PF12	Clears the Natural DBLOG buffer and deactivates logging.

Displaying Adabas Buffers

The Adabas control block is recorded by default. If you want to record one or more Adabas buffers, you need to mark the buffer(s) required in the **DBLOG Menu** before executing the logging function as described in *Specifying Adabas Buffers*. For example, if only logging of the format buffer has been marked in the **DBLOG Menu**, you can only display the **Format Buffer** window but not the **Record Buffer** window.

To display control block or buffer information

1. In the input field next to the required command, enter the line command that corresponds to the required buffer and press ENTER:

Line Command	Requested Buffer
C	Control block
F	Format buffer
R	Record buffer
S	Search buffer
V	Value buffer
I	ISN buffer
.	A period (.) exits the **DBLOG Trace** screen. The current log records are kept in the Natural DBLOG buffer.

A window opens with the log data of the control block or buffer requested. If you entered several line commands, you can press PF3 to view the control block or buffer of the next command.

The following is an example of a window that contains data of a record buffer:

```
16:50:05             ***** NATURAL TEST UTILITIES *****             2008-08-01
User SAG                        - DBLOG Trace -                  Library SAG
M   No Cmd   DB   FNR  Rsp        ISN        ISQ  CID CID(Hex) OP Pgm      Line
_    1 RC    10                                      00000000 F  ATEST    0220
_    2 S1 20000    50              1       2232 ? ?? 02000101    ADATEST  0200
R    3 L1 20000    50              1       2232      40404040    ADATEST  0200
_ +-----------------Page 1 of 1 (logged range:0x-0x4F)-------------------+ 200
_ ! _____  Seq No .. 3     Record Buffer 1/13 (length:0x7A)               ! 340
_ ! 0000 * C1C4D2C9 D5E2D6D5 40404040 40404040 * ADKINSON         * 0000  ! 350
_ ! 0010 * 40404040 0000000B 00000001 40404040 *        ?   ?     * 0010  ! 350
_ ! 0020 * 40404040 00000000 00000000 00000000 *                  * 0020  ! 350
  ! 0030 * 00000000 00000000 00000000 00000000 *                  * 0030  !
  ! 0040 * 00000000 F0F0F0F0 F0F0F0F0 40404040 *     00000000     * 0040  !
  +-------------------------------------------------------------------------+


Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit             --    -     +     ++    <     >     Canc
```

The fields provided in a buffer window are explained in the following table:

Field	Explanation
Page	The number of the current page and the total number of pages generated for the buffer (in the example above, `1 of 1`).
logged range	The buffer length actually logged in hexadecimal format (in the example above, `0x0-0x4F`).

Field	Explanation
Seq No	The sequence number of the command. In the example above, the command was executed in the third place (`3`).
buffer-type *num-current* */num-total*	*buffer-type* denotes the type of buffer requested. In addition, for a format or record buffer, the number of record or format buffers is displayed: *num-current* — Denotes the number of the record/format buffer currently shown. *num-total* — Denotes the total number of the record/format buffers logged. For a database call that uses the extended Adabas control block (ACBX), multiple format/record buffers are logged. The example above shows the first record from a total of 13 records (`1 / 13`). For detailed information on ACBX, see *Adabas Control Block Structures (ACB and ACBX)* in the *Adabas* for mainframes documentation.
length	The total length of the record in hexadecimal format (in the example above, `0x7A`).

Field	Explanation	
_____	In the input field next to **Seq No**, you can enter one of the following line commands:	
	C	Displays the control block.
	F	Displays the format (F) or record (R) buffer.
	or R	If pairs of format and record buffers exist, entering F in a record buffer or R in a format buffer will display the matching record buffer or format buffer respectively. For example, if the second record buffer is currently displayed, entering F will invoke a window with the corresponding second format buffer.
	I	Displays the ISN buffer.
	S	Displays the search buffer.
	V	Displays the value buffer.
	buffer-number	You can enter the number of the record/format buffer you want to view. See also Step 2 below.
	.	A period (.) closes the current buffer window.

2. In a record/format buffer window that contains multiple record/format buffers, you can use one of the following methods to view each record/format buffer:

   Press PF10 to display the previous record/format buffer.

   Or:
   Press PF11 to display the next record/format buffer.

   Or:
   In the _____ input field, enter the number that corresponds to the record/format buffer you want to view.

Displaying Adabas Commands that use Multi-Fetch

If the MULTI-FETCH clause is used in a FIND, READ or HISTOGRAM statement, only the Adabas commands that retrieve a set of records actually access the database. The records retrieved are moved into the multi-fetch buffer from where they are fetched during the execution of the database loop. The next database call is only made for the next set of records. For details, see *Multi-Fetch Clause* in the *Programming Guide*.

The **DBLOG Trace** screen lists both database calls and non-database calls: a database call is marked with an M in the first position of the **OP** column, whereas a non-database call for the multi-fetch buffer is marked with the less-than sign (<). This is demonstrated in the following example.

Example of an Adabas Command with Multi-Fetch

Execute DBLOG for the following Natural program called MFETCH:

```
DEFINE DATA LOCAL
1 EMP-VIEW VIEW OF EMPLOYEES
  2 NAME
END-DEFINE
*
READ (5)  MULTI-FETCH OF 3  EMP-VIEW BY NAME = 'ADKINSON'
  DISPLAY *COUNTER NAME
END-READ
END
```

A **DBLOG Trace** screen similar to the example below appears:

```
10:04:46             ***** NATURAL TEST UTILITIES *****             2008-07-24
User SAG                       - DBLOG Trace -                  Library SAG
M   No Cmd   DB   FNR  Rsp        ISN        ISQ  CID CID(Hex) OP Pgm      Line
_    1 L3    10   316            2108             -?? 00600101 MA MFETCH   0060
_    2 L3    10   316            2109             -?? 00600101 <A MFETCH   0060
_    3 L3    10   316            2110             -?? 00600101 <A MFETCH   0060
_    4 L3    10   316            2111             -?? 00600101 MA MFETCH   0060
_    5 L3    10   316            2112             -?? 00600101 <A MFETCH   0060
_    6 RC    10   316                             -?? 00600101 SI MFETCH   0060
_    7 RC    10                                      00000000 F  MFETCH   0090


Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Print Exit        Posi  --    -     +     ++               Canc
```

The L3 commands listed as sequence numbers 1 and 4 retrieve a set of records from the database (indicated by M in the **OP** column) and return the first record back to the program. The remaining records are cached in the multi-fetch buffer.

The L3 commands listed as sequence numbers 2, 3 and 5 retrieve the record from the multi-fetch buffer (indicated by < in the **OP** column) and return it to the program.

Contents of Record Buffer for Multi-Fetch Database Calls

The record buffer of a database call that uses multi-fetch contains the data of all records retrieved from the database. They are listed in the sequence in which they are processed.

When loading a set of records, Adabas overwrites the record buffer from the first byte to the extent of the records which are returned from the database. Any space left in the buffer is not cleared but still contains data of old records loaded during a previous database call. This means, for example, that if a field defined as `NAME(A20)` is read and a multi-fetch factor of 5 is used, the record buffer has a length of 100 (20 * 5) bytes. If only 3 records are returned from the database, the record buffer is only filled properly with the first 3 records (bytes 1 to 60), whereas the last 2 records (bytes 61 to -100) remain unchanged.

DBLOG Trace Screen for DL/I Calls

- Invoking DBLOG Trace for DL/I Calls
- Screen Columns on DBLOG Trace

Invoking DBLOG Trace for DL/I Calls

The following are example instructions for invoking the **DBLOG Trace** screen for DL/I calls.

1. Write the following Natural program:

```
DEFINE DATA LOCAL
01 COURSE VIEW OF DNDL01-COURSE
  02 COURSEN (A3)
  02 TITLE (A33)
01 OFFERING VIEW OF DNDL01-OFFERING
  02 COURSEN-COURSE (A3)
  02 LOCATION (A31)
END-DEFINE
READ (5) COURSE BY COURSEN
  IF TITLE = 'NATURAL'
    FIND (1) OFFERING WITH COURSEN-COURSE = COURSEN
      MOVE 'DARMSTADT' TO LOCATION
      UPDATE
      END OF TRANSACTION
    END-FIND
  END-IF
END-READ
END
```

2. Enter the following Natural system command:

```
TEST DBLOG D
```

   The message `DBLOG started now` is displayed.

3. Enter the following Natural system command:

```
RUN
```

The Natural program contained in the source area is executed.

4. Enter again:

```
TEST DBLOG D
```

Logging is deactivated and the **DBLOG Trace** screen for DL/I screen is displayed:

```
User SAG                          - DBLOG Trace -                Library SAG
No  Func PCB NS SC DBD/PSB    First SSA (truncated)   IOA (trunc) Program  Line
--- ---- --- -- -- -------- ------------------------ ----------- -------- ----
  1 PCB            PCNQA42                                        LOGDL1   0090
  2 GU   1   1     DNDL01   COURSE  *--(COURSEN =>     .          LOGDL1   0090
  3 GN   1   1     DNDL01   COURSE  *--(COURSEN =>     .Z01       LOGDL1   0090
  4 GN   1   1     DNDL01   COURSE  *--(COURSEN =>     .001       LOGDL1   0090
  5 GN   1   1     DNDL01   COURSE  *--(COURSEN =>     .004NATURA LOGDL1   0090
  6 GHNP 1   2     DNDL01   COURSE  *- (COURSEN  =004  ?010791DAR LOGDL1   0110
  7 REPL 1         DNDL01                              ?010791DAR LOGDL1   0130
  8 SYNC                                                          LOGDL1   0140
  9 PCB            PCNQA42                                        LOGDL1   0110
 10 GU   1   1     DNDL01   COURSE  *--(COURSEN = 004  .004NATURA LOGDL1   0110
 11 GHNP 1   2     DNDL01   COURSE  *--(COURSEN = 004  ?010791DAR LOGDL1   0110
 12 GN   1   1     DNDL01   COURSE  *--(COURSEN =>     +110       LOGDL1   0090
***** End of Log *****




NEXT                                                             LIB=SAG
```

Screen Columns on DBLOG Trace

The columns of fields provided on the **DBLOG Trace** screen for DL/I calls are described in the following section.

Column	Explanation
No	Sequence number. The calls are displayed in the sequence in which they were executed.
Func	DL/I function.
PCB	PCB number.
NS	Number of SSAs.
SC	DL/I status code.
DBD/PSB	DBD name for DB calls. PSB name for scheduling calls.
First SSA	First 25 bytes of the first SSA.
IOA	First 13 bytes of the I/O area.
Program	Natural program name.
Line	Source-code line number.

DBLOG Trace Screen for SQL Statements

- Invoking DBLOG Trace for SQL Statements
- Screen Columns and Commands on DBLOG Trace

Invoking DBLOG Trace for SQL Statements

The following is an example of invoking the **DBLOG Trace** screen for SQL statements.

1. Write the following Natural program:

```
DEFINE DATA LOCAL
01 EMP VIEW OF DSN8810-EMP
  02 EMPNO
  02 FIRSTNME
  02 MIDINIT
  02 LASTNAME
  02 EDLEVEL
  02 SALARY
01 EMPPROJACT VIEW OF DSN8810-EMPPROJACT
  02 EMPNO
  02 PROJNO
  02 ACTNO
  02 EMPTIME
END-DEFINE
FIND (1) EMP WITH EMPNO > '000300'
  FIND (1) EMPPROJACT WITH EMPNO = EMPNO(0150)
    MOVE 0.75 TO EMPTIME
    UPDATE
  END-FIND
  ADD 1 TO EDLEVEL
  UPDATE
END-FIND
*
FIND (1) EMP WITH EMPNO > '000300'
  FIND (1) EMPPROJACT WITH EMPNO = EMPNO(0240)
```

```
    DISPLAY EMPPROJACT EMP.EDLEVEL
  END-FIND
END-FIND
ROLLBACK
END
```

2. Enter the following Natural system command:

```
TEST DBLOG Q
```

   The message `DBLOG started now` is displayed.

3. Enter the following Natural system command:

```
RUN
```

   The Natural program in the source area is executed.

4. Enter again:

```
TEST DBLOG Q
```

   Logging is deactivated and a **DBLOG Trace** screen for SQL statements similar to the example below appears:

```
11:28:58              ***** NATURAL Test Utilities *****             2008-07-28
User SAG                         - DBLOG Trace -                Library SAG
M No   R SQL Statement (truncated)     CU SN SREF M Typ SQLC/W Program  Line LV
_    1   SELECT EMPNO,FIRSTNME,MIDINIT 01 01 0150 D DB2        LOGSQL   0150 01
_    2     FETCH CURSOR NEX            01 01 0150 D DB2        LOGSQL   0150 01
_    3   SELECT EMPNO,PROJNO,ACTNO,EMP 02 02 0160 D DB2        LOGSQL   0160 01
_    4     FETCH CURSOR NEX            02 02 0160 D DB2        LOGSQL   0160 01
_    5   UPDATE DSN8810.EMPPROJACT SET 02 03 0160 D DB2        LOGSQL   0180 01
_    6   CLOSE CURSOR                  02 02 0160 D DB2        LOGSQL   0160 01
_    7   UPDATE DSN8810.EMP SET EDLEVE 01 04 0150 D DB2        LOGSQL   0210 01
_    8   CLOSE CURSOR                  01 01 0150 D DB2        LOGSQL   0150 01
_    9   SELECT EMPNO,FIRSTNME,MIDINIT 05 05 0240 D DB2        LOGSQL   0240 01
_   10     FETCH CURSOR NEX            05 05 0240 D DB2        LOGSQL   0240 01
_   11   SELECT EMPNO,PROJNO,ACTNO,EMP 06 06 0250 D DB2        LOGSQL   0250 01
_   12     FETCH CURSOR NEX            06 06 0250 D DB2        LOGSQL   0250 01
_   13   CLOSE CURSOR                  06 06 0250 D DB2        LOGSQL   0250 01
_   14   CLOSE CURSOR                  05 05 0240 D DB2        LOGSQL   0240 01
_   15   ROLLBACK                      00 00 0000 D DB2        LOGSQL   0290 01
_
_
Command ===>


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Print Exit  Top   Posi  Bot    -     +                     Canc
```

Screen Columns and Commands on DBLOG Trace

The columns of fields and commands provided on the **DBLOG Trace** screen for SQL statements are described in the following section. You execute a command by either pressing a PF key or entering a direct command in the Command line.

Column	PF Key Direct Command	Explanation
M		Input option for line commands: `E` Executes the `EXPLAIN` command which provides information on the DB2 or SQL/DS optimizer's choice of strategy for executing SQL statements. See also *DB2 EXPLAIN Command* in the *Natural for DB2* documentation and *EXPLAIN Command* in the *Natural for SQL/DS* documentation. `L` Executes the `LISTSQL` command which lists the Natural statements in the source code of an object and the corresponding SQL statements into which they have been translated. An SQL statement is identified by the library name, program name, and line number taken from the Natural DBLOG buffer. See also *LISTSQL Command* in the *Natural for DB2* documentation and *LISTSQL Command* in the *Natural for SQL/DS* documentation. **Important:** Since both commands obtain their information from the Natural system file, unwanted results may occur if the corresponding Natural program has been recataloged after the logging function was executed with the `TEST DBLOG Q` command. These unwanted results may be caused by statements modified after the logging.
No		Sequence number; the statements are displayed in the sequence in which they were executed.

Column	PF Key Direct Command	Explanation
R		Only applicable if the Natural file server for DB2 is in use. Indicates by an asterisk in front of the corresponding statement that a reselection has been performed; if not, the column is left blank. See also *Concept of the File Server* in the *Natural for DB2* documentation.
SQL Statement		The first 29 characters of the logged SQL statement.
CU		Cursor number.
SN		Internal statement number.
SREF		Statement reference number.
M		Mode: `D` for dynamic or `S` for static.
Typ		Database type: DB2 or /DS.
SQLC/W		Either the SQL return code in the SQLCODE field of the `SQLCA`, or the warning in the SQLWARN0 field of the `SQLCA` if `SQLCODE` is `0`.
Pgm		Natural program name.
Line		Source code line number.
LV		Program level.
	PF2	Prints a hardcopy of a screen shot.
	PF3	Exits the **DBLOG Trace**. The current log records are kept in the Natural DBLOG buffer.
	PF4	Scrolls to the beginning of the list.
	PF5	Moves log entries to the top of the screen: In column M, position the cursor next to the desired command and sequence number listed in column No and choose PF5. The logs are repositioned starting with the sequence number selected.
	PF6	Scrolls to the end of the list.
	PF7 or -	Scrolls up one page in the list.
	PF8 or +	Scrolls down one page in a list.
	PF12	Clears the Natural DBLOG buffer and deactivates logging.