

Adabas Dump Formatting Tool (ADAFDP)

This section describes the use of the Adabas dump formatting tool ADAFDP.

- ADAFDP Function
 - ADAFDP Output
-

ADAFDP Function

ADAFDP is the address space dump formatting module. During abnormal shutdown of the Adabas nucleus, this module receives control to format and display information that should help you analyze the reason for the error.

During a nucleus shutdown, ADAMPM determines the shutdown reason. If the reason is abnormal termination, ADAMPM loads the ADAFDP module into the address space prior to the 20 call. ADAFDP subsequently receives control to format nucleus information.

If ADAFDP cannot be loaded, message ADAF03 is written to the console and abnormal shutdown continues.

ADAFDP Output

Much of the information formatted by ADAFDP is self-explanatory. However, because the type and amount of information depends on the shutdown situation, a summary of ADAFDP output is provided in this section.

- ADAFDP Messages
- Pool Abbreviations
- User Threads
- Command Information
- RABN Information

ADAFDP Messages

| Message | Description |
|---------------------------|---|
| ADAH51 / ADAH52 | The message is displayed on the console and written to DDPRINT at the point where the format begins and terminates. |
| ADAMPM ABEND CODE and PSW | If an abend code and program status word (PSW) were saved in ADAMPM by the Adabas ESTAE, ADAFDP displays these. In addition, ADAFDP determines the module whose entry point best fits the PSW and calculates the offset within that module. If the ADAMPM abend code and PSW are zero, ADAFDP does not format this information. |

| Message | Description |
|--|---|
| ADABAS MODULE LOCATIONS | ADAFDP formats and displays the location of each of the Adabas nucleus modules resident in the address space. |
| ADDRESS LOCATIONS FOR USER EXITS | ADAFDP formats and displays the location of any user exit loaded with the Adabas nucleus. |
| ADDRESS LOCATIONS FOR HYPEREXITS | ADAFDP formats and displays the location of any hyperdescriptor exit loaded with the Adabas nucleus. Hyperdescriptor exits 10-31 are displayed as A-U, respectively. |
| ADANC0 STANDARD REGISTER SAVE AREA | Registers 0-7/8-F, which are saved in ADANC0. ADAFDP determines if any of these registers contains an address that points at a nucleus pool in storage. If yes, ADAFDP indicates which pool and snaps storage at that address. If the register is 12 and it points to a user thread, ADAFDP snaps the entire thread. |
| ADANC0 ABEND SAVE REGISTERS | Registers 0-7/8-F, which are saved in ADANC0 as a result of a user abend. ADAFDP determines if any of these saved registers contains an address that points at a nucleus pool in storage. If yes, ADAFDP indicates which pool and snaps storage at that location. If the saved register is 12 and it points to a user thread, ADAFDP snaps the entire thread. |
| ADAMPM SAVE REGISTERS | Registers 0-7/8-F, which were saved in ADAMPM by the Adabas ESTAE. These are the same registers displayed with the ADAM99 message. ADAFDP determines if any of these saved registers contains an address that points within a nucleus pool in storage. If yes, ADAFDP indicates which pool and snaps storage at that location. |
| BEGIN / ENDING ADDRESSES OF POOLS / TABLES | ADAFDP determines begin/ending address locations for pools and tables for the Adabas nucleus. These addresses are presented for easy location in the actual dump. See <i>Pool Abbreviations</i> for more information. |
| ADABAS THREADS | ADAFDP formats the physical threads including threads 0, -1, and -2. The number of lines depends on the value of NT. The thread that was active at the time of the abnormal termination (if any) is marked by a pointer “- ->”. |
| USER THREADS | For any of the threads -2 to NT that had assigned work to perform, ADAFDP formats and displays information about the status of that thread. See <i>User Threads</i> for more information: |
| FOLLOWING COMMANDS WERE FOUND IN THE CMD QUEUE | ADAFDP scans the command queue and formats information for any command found in the queue. See <i>Command Information</i> for more information. |
| POOL INTEGRITY CHECK | ADAFDP check the integrity of several pools within the Adabas nucleus address space. If an error is detected within that pool, ADAFDP indicates which pool and what type of error was encountered. In addition, ADAFDP snaps storage at the location where the error was detected. |
| FOLLOWING RABNS / FILES ACTIVE IN BUFFER POOL | ADAFDP scans the buffer pool header for RABNs that were active or being updated. See <i>RABN Information</i> for more information. |

| Message | Description |
|--|--|
| ADAIOR REGS FOUND AT OFFSET X'080' | Registers 0-7/8-F found saved in ADAIOR at this offset. If ADAFDP determines that any of these register values is pointing within an Adabas pool, it snaps storage at that location. |
| ADAIOR REGS FOUND AT OFFSET X'0C0' | Registers 0-7/8-F found saved in ADAIOR at this offset. If ADAFDP determines that any of these register values is pointing within an Adabas pool, it snaps storage at that location. |
| ICCB POINTED FROM X'A0' IN IOR | The ICCB address to which this offset in ADAIOR points. |
| ADAI22 ADAIOR TRACE TABLE | Format of ADAIOR trace table; same as that found with the ADAM99 message. |

Pool Abbreviations

| Pool Abbreviation | Description |
|-------------------|--|
| LOG | Log area |
| OPR | Adabas nucleus operator command processing area |
| CQ | Address of the command queue, which is formatted later by ADAFDP |
| ICQ | Internal command queue |
| TT | Thread table |
| IA1 | Software AG internal area 1 |
| SFT | Session file table |
| FU | File usage table |
| FUP | File update table |
| IOT | I/O table for asynchronous buffer flushing |
| PL2 | PLOG area for asynchronous buffer flushing |
| PET | Table of posted ETs |
| TPT | Tpost |
| TPL | Tplatz |
| UQP | Unique descriptor pool |
| UHQ | Upper hold queue |
| HQ | Hold queue |
| UUQ | Upper user queue |
| UQ | User queue |
| FP | Format pool |

| Pool Abbreviation | Description |
|--------------------------|--|
| FHF | File HILF element |
| PA | Protection area |
| TBI | Table of ISNs |
| TBQ | Table of sequential searches |
| WK3 | Work part 3 space allocation table |
| IA2 | Software AG internal area 2 |
| WK2 | Work part 2 space allocation table |
| VOL | VOLSER table |
| WIO | Work block I/O area |
| FST | Free space table work area |
| UT | User threads |
| WP | Work pool |
| AW2 | Work block asynchronous I/O area |
| IOP | I/O pool related to asynchronous buffer flush |
| IU2 | Buffer pool importance header upper 2 |
| IU1 | Buffer pool importance header upper 1 |
| BU2 | Buffer pool upper header 2 |
| BU1 | Buffer pool upper header 1 |
| BH | Address location of the buffer pool header, information from the buffer pool header is formatted later by ADAFDP |
| BP | Address location of the physical start of the buffer pool |

User Threads

| Information | Description |
|--------------------|--------------------|
| Thread Number | -2 to NT |

| Information | Description |
|-------------------|--|
| Status | <p>Indicates the current status of the thread. The following statuses are possible:</p> <ul style="list-style-type: none"> ● *Active*: the currently active thread ● In Use: thread has been assigned work ● Waiting For I/O: waiting for a block not in buffer pool ● Waiting For RABN: waiting for a RABN already in use ● Waiting For Work-2 Area Block: similar to waiting for I/O ● Waiting Workpool Space: provides number of bytes in decimal ● Ready To Run: waiting to be selected for execution |
| CMD | The Adabas command being executed |
| Response Code | Response code (if any) |
| File Number | File number for this command |
| ISN | Internal sequence number for this command |
| Sub. Rsp | Subroutine response code (if any) |
| Last RABN for I/O | Last RABN required by command processing, in decimal |
| Type | Last RABN type (A - ASSO, D - DATA) |
| CQE Addr | Command queue element address for this command |
| User Jobname | Job name for user who executed this command |
| ITID | Internal Adabas ID for user who executed this command |
| User | User ID for user who executed this command |
| Unique global ID | 28-byte ID for user who owns this command |
| Buffer Addresses | buffer addresses for: control block, format buffer, search buffer, value buffer, ISN buffer |
| Buffer Lengths | FL: format buffer length RL: record buffer length SL: search buffer length VL: value buffer length IL: ISN buffer length |
| Snap Thread | The first 144 bytes of the user thread are snapped |

Command Information

| Information | Description |
|----------------|--|
| CQE Address | The address location of this CQE |
| F | <p>Command queue flag bytes:</p> <ul style="list-style-type: none"> ● First Byte: General Purpose Flag <ul style="list-style-type: none"> ○ X'80': User buffers in service partition, region, address space ○ X'40': ET command waiting for 12 call ○ X'20': Waiting for 16 call ○ X'10': 16 call required ○ X'08': Attached buffer ○ X'04': Attached buffer required ○ X'02': X-memory lock held (z/OS only) ● Second Byte: Selection Flag <ul style="list-style-type: none"> ○ X'80': In process ○ X'40': Ready to be selected ○ X'20': Search for UQE done ○ X'10': UQE found ○ X'08': Not selectable during BSS=x'80' status ○ X'04': Not selectable during ET-SYNC ○ X'02': Waiting for space ○ X'01': Waiting for ISN in HQ |
| CMD | The command type |
| File Number | The file number for this command |
| Job Name | Job name for the user |
| Addr User | UQE Address of users UQE, if searched for and found |
| Addr User ASCB | Address location of user's ASCB |
| Addr ECB | Address location of user's ECB (in user's address space) |
| Addr User UB | Address of users UB (in user's address space) |

| Information | Description |
|------------------|---|
| Addr User PAL | Address location of user's parameter address list |
| CQE ACA | ACA field of CQE. |
| CQE RQST | RQST field of CQE |
| Abuf/Pal | Address of the attached buffer/parameter address list (PAL) for CMD |
| Comm Id | 28-byte unique user ID for this command |

RABN Information

| Information | Description |
|-------------|--|
| RABN Number | The RABN number in decimal |
| Type | Type of block (A - ASSO, D - DATA) |
| Flag | BP header element flag byte: <ul style="list-style-type: none"> ● AKZ X'40': Active indicator ● UKZ X'20': Update indicator ● RKZ X'10': Read indicator ● XKZ X'04': Access is waiting for block ● YKZ X'02': Update is waiting for block ● SKZ X'01': Write indicator |
| File | File number that owns this block |
| Address | Address location of block in storage. |