

ACCOUNTING BLOCK SERVICES

4/22/71

All manipulations of Accounting Blocks are to be performed using a single operation (DF:ABSV in OPERCL). The 1st parameter indicates the service to be performed. Unneeded parameters are indicated by "-" and are ignored. A maximum of 8 data words may be returned to the caller; no capabilities are ever returned.

All errors are class E.ABLOCK unless otherwise noted.

4/22/71

Service: Create Accounting Block

Input Parameters:

IP1 D: 0
IP2 D: -
IP3 D: -
IP4 D: -

Output Parameters:

D: (1 word) Accounting Block Number of new block
C: (none)

Action: New accounting block is created in Inactive state.
All reservations, charges, etc. are zero, and may not
be changed until the block has been activated.

Errors: (none)

4/22/71

Service: Activate Accounting Block

Input Parameters:

IP1 D: 1
IP2 D: Accounting Block Number
IP3 D: -
IP4 D: -

Output Parameters:

D: (none)
C: (none)

Action: Places the Accounting Block in the active state, allowing space to be moved in, charges to accumulate, etc.

Errors:

E.NOABLK Accounting Block does not exist.
E.ACTIV Accounting Block already active.

4/22/71

Service: Destroy Accounting Block

Input Parameters:

IP1 D: 2
IP2 D: Accounting Block Number
IP3 D: -
IP4 D: -

Output Parameters:

D: (none)
C: (none)

Action: Accounting Block is destroyed unless destruction is prevented by some error condition.

Errors:

E.NOABLK Accounting Block does not exist, or is inactive.
E.BUSY Accounting Block still funding processes and/or files.
E.RESV Accounting Block still holding reserved space.
E.BUSY Accounting Block "Account Tag" field non-zero.
E.BUSY Charge meters in Accounting block non-zero.

4/22/71

Service: Display Accounting Block

Input Parameters:

IP1 D: 3
IP2 D: Accounting Block Number
IP3 D: -
IP4 D: -

Output Parameters:

D: (8 words) Contents of Accounting Block
C: (none)

Action: The entire accounting block is returned.

Errors:

E.NOABLK Accounting Block does not exist, or is inactive.

4/22/71

Service: Move Disk Space (Father-to-Son)

Input Parameters:

IP1 D: 4
 IP2 D: Accounting Block Number (Father)
 IP3 D: Accounting Block Number (Son)
 IP4 D: Amount of Disk Space (sectors)

Output Parameters:

D: (none)
 C: (none)

Action: Surplus := Reserved(Father) - Occupied(Father);
if Surplus < IP4 then go to Trimfather;
 Occupied (Father) := Occupied(Father) + IP4;
 Reserved (son) := Reserved (son) + IP4;
 Rate(son) := Rate(son) + IP4;
 Return;
 Trimfather: Occupied(Father) := Reserved(Father);
 Reserved(Son) := Reserved(son) + Surplus;
 Rate(Son) := Rate(son) + Surplus;
 X6(Caller) := Surplus;
 Freturn

Errors:

E.NOABLK Accounting block does not exist
 E.NEGPAR* Amount to move is negative

(* class = E.PARMS)

4/22/71

Service: Move Disk Space (Son-to-Father)

Input Parameters:

IP1 D: 5
IP2 D: Accounting Block Number (Father)
IP3 D: Accounting Block Number (Son)
IP4 D: Amount of Disk Space (sectors)

Output Parameters:

D: (none)
C: (none)

Action: Surplus := Reserved(Son) - Occupied(Son);
if Surplus < IP4 then go to Trimson;
Occupied (Father) := Occupied(Father) - IP4;
Reserved (Son) := Reserved (Son) - IP4;
Rate(Son) := Rate(son) - IP4;
Return;
Trimson: Occupied(Father) := Occupied(Father) - Surplus;
Reserved(Son) := Occupied(son);
Rate(Son) := Rate(Son) - Surplus;
X6(Caller) := Surplus;
Freturn

Errors:

E.NOABLK Accounting Block does not exist
E.NEGPAR* Amount to move is negative

(* class = E.PARMS)

5/4 /71

Service: Increment Charge Meter

Input Parameters:

IP1 D: 6
IP2 D: Accounting block Number
IP3 D: Increment (+ or -)
IP4 D: -

Output Parameters:

D: (none)
C: (none)

Action: The increment is added to the discontinuous charge meter in the accounting block. (60 bit integer add). The meter counts up by usec/1024 , hence the units of the increment should be sector-milliseconds.

Errors:

E.NOABLK Accounting block does not exist (or is inactive)

4/22/71

Service: Set Accounting Tag in Accounting Block

Input Parameters:

IP1 D: 7
IP2 D: Accounting Block Number
IP3 D: Value for Tag
IP4 D: -

Output Parameters:

D: (none)
C: (none)

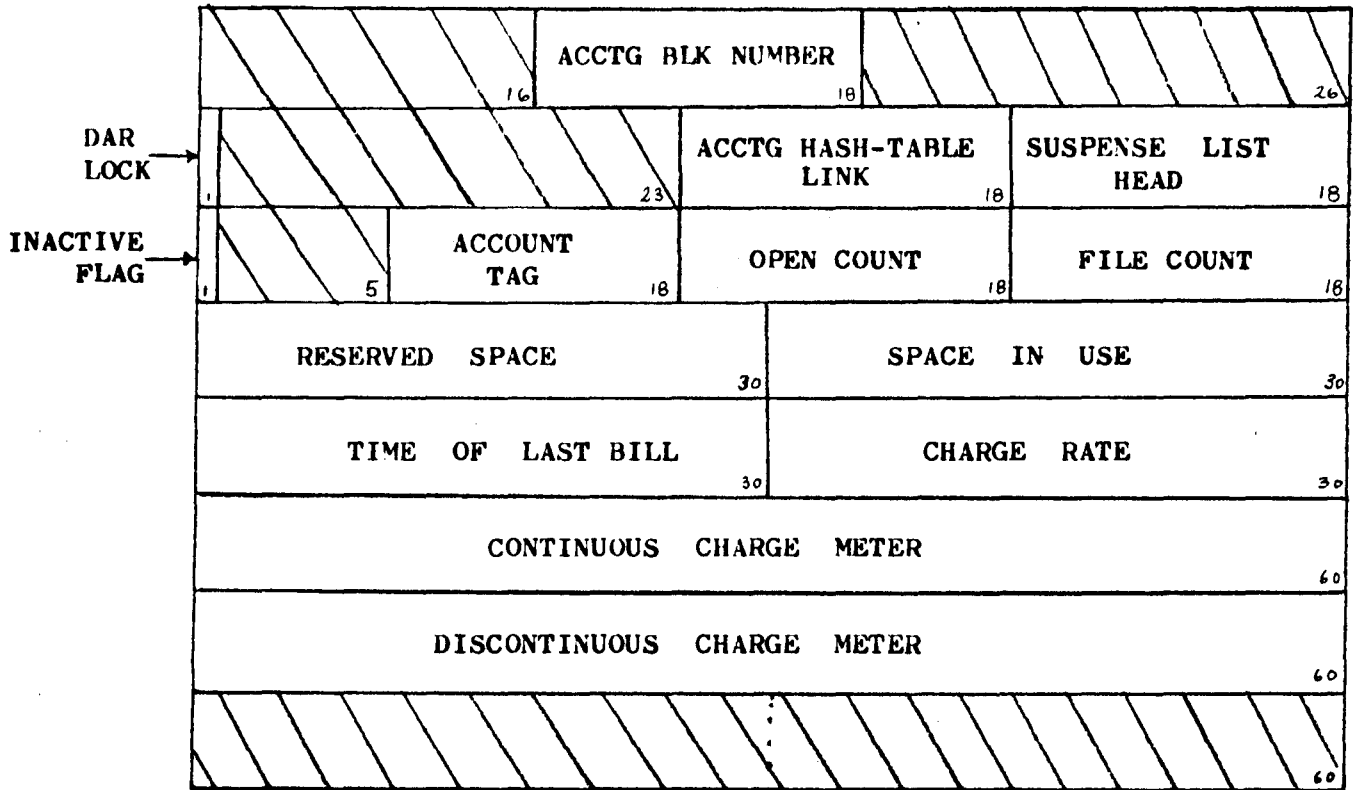
Action: Sets the 18 bit Account Tag field in the Accounting Block to the value provided.

Errors:

E.NOABLK Accounting Block does not exist, or is inactive.

Paul

DAR (Disk Accounting Record)



FIELDS IN ACCOUNTING BLOCK:

1. Accounting block number: identifies this block; used to hash in DAR hash table of open accounting blocks.
- * 2. Lock: Disk system internal lock bit.
3. Account Tag: settable 18-bit field.
- * 4. Hash table link: link in collision chain in DAR hash table.
- * 5. Suspense list head: head of list of waiting processes.
- * 6. Time of last charge meter update: The time in milliseconds (usec/1024) between the last deadstart and the last update of charge meters in this accounting block.
7. Open Count: Sum of 1) no. of processes funding swapped ECS from this accounting block, and
2) no. of files funding disk space from this accounting block which are currently open.
8. No. of files funded: Total files (open or closed) which fund their disk space from this accounting block.
9. Swapped ECS Charge Rate: Rate at which Swapped ECS charge meter is counted up each millisecond.
10. (Disk) Charge rate: Rate at which Disk charge meter is counted up each millisecond.
11. Swapped ECS reserved: ECS available for open files of processes funded from this accounting block. (units = words)
12. (Disk) Reserved: Disk space available for files funded from this accounting block (units = sectors).
13. Swapped ECS occupied: Total swapped ECS occupied by open files of processes funded from this accounting block.
14. Disk Occupied: Total disk space occupied by files funded from this accounting block.
15. Swapped ECS charge meter: Total charge in word-milliseconds accumulated in this accounting block.
16. Disk Charge meter: Total charge in sector-milliseconds accumulated in this accounting block.

17. *Discipline charge meter*

* Normally not of interest outside disk system

Creation and Destruction of Accounting Blocks

Accounting blocks are created and destroyed only by the directory system during creation or destruction of a funding directory. Three calls on the Low Level Disk system are used:

1) Create Accounting Block

Creates a new accounting block in an inactive state and returns the accounting block number as a return parameter.

2) Activate Accounting Block (Acctg Blk #)

Activates an inactive accounting block. No other actions may be performed on an inactive accounting block (i.e. space may not be moved in, etc.)

3) Destroy Accounting Block (Acctg Blk #)

Destroys the accounting block. The block may not be holding any resources and the charge meters must read zero.

It is important that, regardless of system crashes, each existing accounting block must be pointed to by a funding directory, since otherwise, it and all its resources are permanently lost. The directory system must therefore observe the following conventions when creating or destroying a funding directory.

Creating a Funding Directory:

1. Create new funding directory and fabricate ownership entry in its father directory.
2. Pseudo-close father directory to assure that the ownership entry appears on the disk.
3. Create a new accounting block. Store the number of the accounting block in the new directory.
4. Pseudo-close the new directory to assure that the accounting block pointer appears on the disk.
5. Activate the accounting block.

Destroying a Funding Directory:

1. Pick up the accounting block number from the directory. Destroy accounting block (if error due to resources, charges, etc., pass back to caller.)
2. Destroy the directory.

Remarks:

Note that the above procedure, while never allowing an accounting block to get lost, does allow the directory system to find directories with invalid accounting block numbers (causes error when passed to Low Level disk). This should be taken to mean that the system crashed halfway through creation or destruction of a funding directory; probably, the directory should be destroyed. (it cannot contain any entries.)

IDENT DABUF
TITLE DISK ACCOUNTING BLOCK/RECORD BUFFER

*
ENTRY DA.LEN,DA.LOG

* DAR/DAB BUFFER AREA

Paul McJones

*
L MACRO LABEL,L
BSS 0
ENTRY L
ENDM

Note: DR:DSRK = destroy disk object

*
DAB LABEL
DAR LABEL

*
DA.ABNO LABEL ACCOUNTING BLOCK NUMBER
BSS 1

*
DA.FREE LABEL FREE/BUSY FLAG AND CHAIN LINK IN DAB
DA.LOCK LABEL LOCK BIT IN DAR
DA.TAG LABEL WRITE-ABLE ACCOUNT TAG
DA.HASH LABEL HASH CHAIN LINK IN DAR
DA.SUSP LABEL SUSPENSE LIST HEAD IN DAR
BSS 1

*
DA.TIME LABEL TIME OF LAST CHARGE-METER UPDATE
DA.OPEN LABEL OPEN COUNT
DA.NFIL LABEL NO. OF FILES FUNDED
BSS 1

*
DA.RATE LABEL SWAPPED ECS / DISK CHARGE RATES
BSS 1

*
DA.RESV LABEL SWAPPED ECS / DISK SPACE RESERVED
BSS 1

*
DA.OCCU LABEL SWAPPED ECS / DISK SPACE OCCUPIED
BSS 1

*
DA.SCHG LABEL SWAPPED ECS CHARGE
BSS 1

*
DA.DCHG LABEL DISK CHARGE
BSS 1

*
*
*
*
*
DA.LEN EQU *-DAB
DA.LOG EQU 3 ***** NOTE: MUST EQUAL LOG2 OF DA.LEN *****

*
DARADDR ENTRY DARADDR
BSS 1 DDS ADDRESS OF DAR

*
*
END