TurboIMAGE INTERNAL FILE STRUCTURE

Doris Chen
Hewlett-Packard
Cupertino, California, USA

Introduction

Through the last decade, many people have developed interest and expertise in IMAGE. Thousands of programs and products have been written using IMAGE databases. With the release of TurboIMAGE, an enhanced version of IMAGE, the internal file structure has changed. To upgrade to TurboIMAGE, databases must be converted. The purpose of this paper is to provide IMAGE experts a better understanding of the new file structure. Any privileged mode programs dependent on IMAGE internal file structures may require modifications to run with TurboIMAGE. The complete TurboIMAGE file structure is presented here. The differences from IMAGE are discussed. Some examples will be given during the live presentation. Note: Some information in this paper will also be included in the U-MIT System Tables Reference Manual.

TurboIMAGE vs IMAGE

TurboIMAGE has relaxed many IMAGE limitations (refer to the TurboIMAGE Reference Manual for the new limitations). These are the reasons for changing the database data structure. A TurboIMAGE database consists of 3 types of files: a root file, master data sets, and/or detail data sets (same as IMAGE). Master data sets must be converted due to the change of the detail chain count in the master set entries. It was increased from a single word (65K entries in a detail chain) to a double word. The root file must be converted due to the following changes:

a. The maximum number of items in a database was increased from 255 to 1023. An item number can no longer be stored in a byte, it requires a word. Thus, the Item Table and Data Set Control Blocks were changed.

b. The maximum number of data sets in a database was increased from 99 to 199. The Set Table and Data Set Control Blocks were changed to accommodate additional data sets in the database and additional fields in the data set. (A field is an item defined in the data set).

c. All major tables, i.e. the Item Table, the Set Table, and Data Set Control Blocks, have been moved to begin at the record boundry for easy access.

d. Some information has been added to the root file information (label 0) and the database global information (record 0) for general housekeeping purposes.

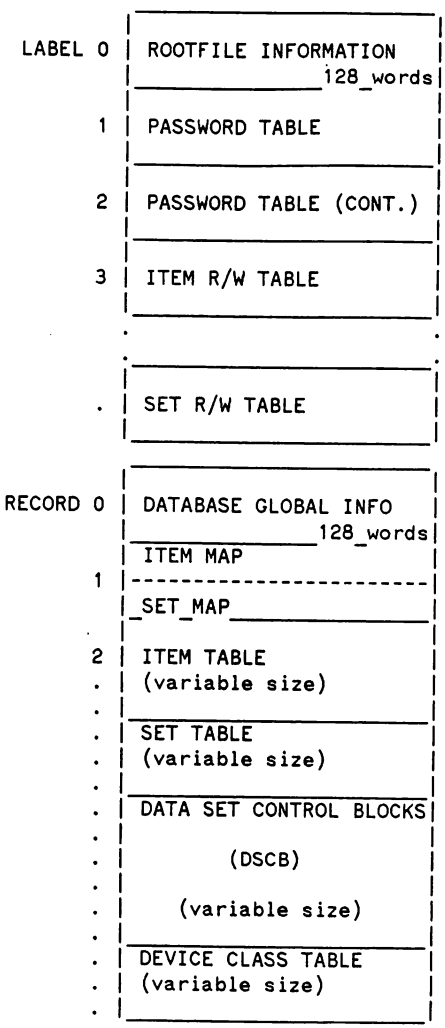
e. Two tables were added: Item/Set Maps and the Device Class Table.

Throughout this paper, a '+' next to a variable indicates the value or the meaning of the variable is new or has been changed for TurboIMAGE.

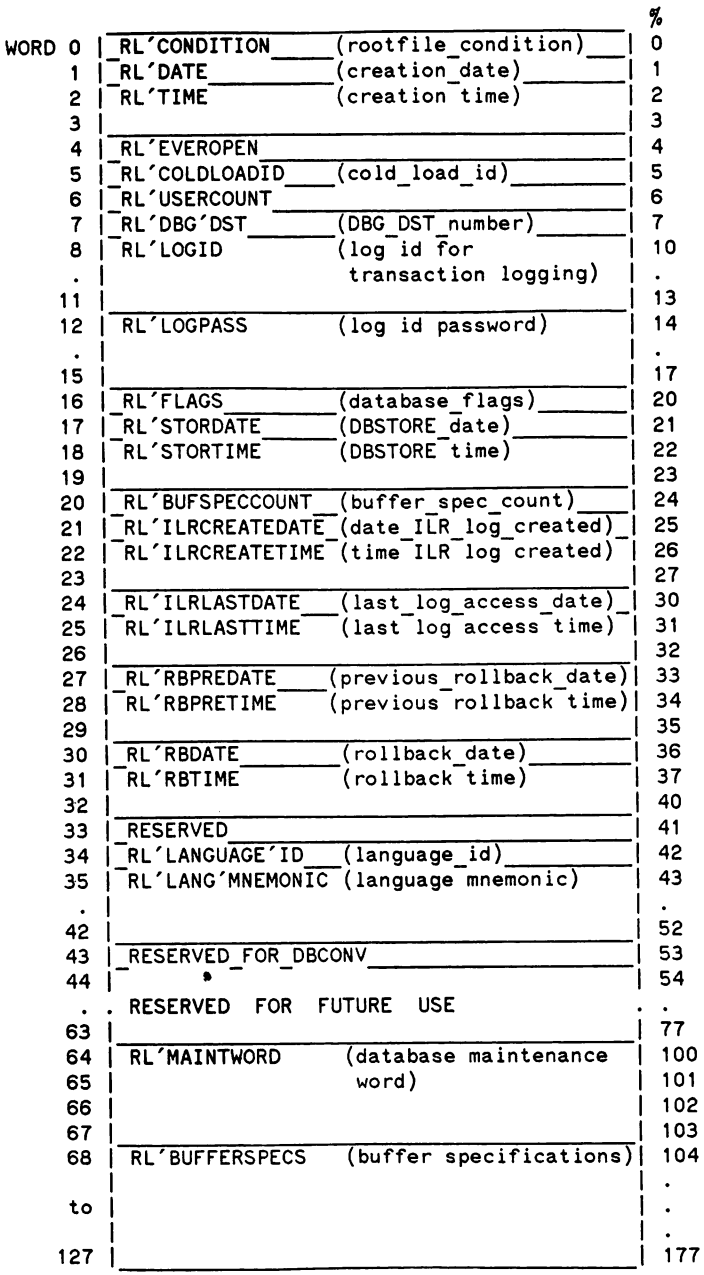
Root File Structure

The database root file is an MPE file, with a file code of -400, created by DBSCHEMA. It consists of information about the database: the password table, item table, set table, etc. The record size is 128 words, fixed, binary format with a file system blocking factor of 1. The size of the file depends on the number of data items and data sets defined in the database.

Following is the general format of the root file. The detailed format of each table in the root file follows the general format.



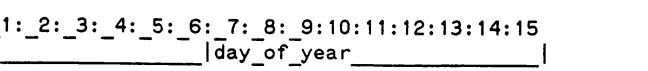
Root File Information (Label 0)



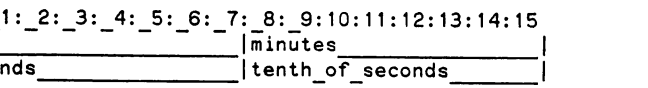
RL'CONDITION (IN ASCII):
TurboIMAGE uses 2 ASCII characters to indicate the condition of the database. Following are the condition codes:

JB		- The database has not been created yet.
FW		- The database is OK.
RM		- Random Modification. The database is being modified with output defer mode.
MC		- Maintenance Create. The database is being created by DBUTIL.
ME		- Maintenance Erase. The database is being erased by DBUTIL.
IL		- ILR is in progress (DBOPEN).
IE		- ILR enable is in progress (DBUTIL).
ID		- ILR disable is in progress (DBUTIL).
CN	(+)	- Conversion by DBCONV is in progress. The process can NOT be continued if DBCONV is interrupted.
CA	(+)	- Conversion by DBCONV is in progress. The process can be continued if DBCONV is interrupted.
MV	(+)	- Data set move is in progress (DBUTIL).

RL'DATE: Root file creation date*. Its format is:



RL'TIME: Root file creation time*. Its format is:



RL'EVEROPEN: This field is no longer used under IMAGE B and TurboIMAGE.

RL'COLDLOADID: MPE system cold load id. This id and the RL'USERCOUNT are used to determine if there was a system failure while the database was in used.

RL'USERCOUNT: The number of users currently accessing the database.

RL'DBG'DST (+): The DBG (Data Base Global Control Block) DST number. When the database is opened by the first user, this number is stored here. Subsequent users can find the database DBG easily.

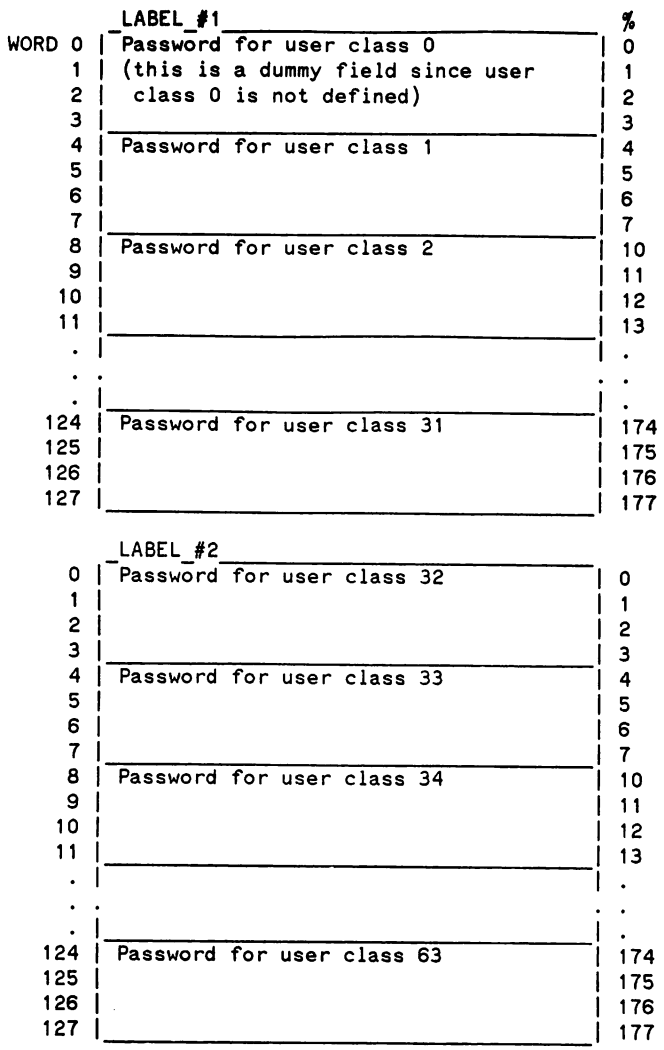
RL'FLAGS:

bit 0	- RECOVERY	Default is NO (0)
1	- LOGGING	Default is NO (0)
2	- ACCESS	Default is YES (1)
3	- DUMPING	Default is NO (0)
4	- AUTO DEFER (+)	Default is NO (0)
5-6	- SUBSYSTEM ACCESS	Default is R/W (00)

```
          7 - ILR                   Default is NO  (0)
          8 - ROLLBACK              Default is NO  (0)
          9 - Reserved for future use.
         10 - DIRTY FLAG            Default is YES (1).
                                    This indicates the database has
                                    been modified but not DBSTOREd.
         11 - DBRECOV RESTART (+) Default is NO  (0)
       12-15 - Reserved for future use.

       Bits 0 to 8 are set by DBUTIL.  Bit 10 is set by DBSTORE, DBDELETE,
       DBPUT, or DBUPDATE.  Bit 11 is set by DBRECOV.

RL'STORDATE: Same format as RL'DATE*.

RL'STORTIME: Same format as RL'TIME*.

RL'BUFSPECCOUNT: Maximum number of buffer specifications allowed.

RL'ILRCREATEDATE: Same format as RL'DATE*.

RL'ILRCREATETIME: Same format as RL'TIME*.

RL'ILRLASTDATE: Same format as RL'DATE*.

RL'ILRLASTTIME: Same format as RL'TIME*.

RL'RBPREDATE (+): Same format as RL'DATE*.

RL'RBPRETIME (+): Same format as RL'TIME*.

RL'RBDATE (+): Same format as RL'DATE*.

RL'RBTIME (+): Same format as RL'TIME*.

RL'LANGUAGE'ID: Same format as defined in the system configuration.

RL'LANG'MNEMONIC: Language mnemonic for this database.
                  Maximum of 16 characters.

RL'MAINTWORD: For data bases with no maintenance word this field has
              2 semicolons (';;') and trailing blanks.
RL'BUFFSPECS:
              0: 1: 2: 3: 4: 5: 6: 7: 8: 9:10:11:12:13:14:15   %
   WD    68 |buffers for 1 user     |buffers for 2 users     | 104
         69 |buffers for 3 users     |buffers for 4 users     | 105
          . | etc...                                          | .
          . |_________________________________________________| .
        127 |buffers for 119 users   |buffers for 120 users   | 177
```

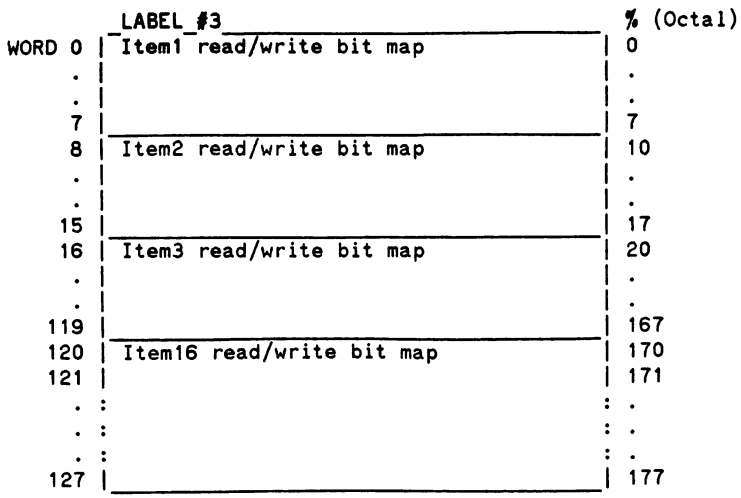
* All DATE and TIME fields can be formatted (for display purposes) individually by calling the FMTCALENDAR and FMTCLOCK Intrinsics respectively. Both fields can also be formatted at once with the FMTDATE Intrinsic.

Password Table



The Password Table occupies user labels number 1 and 2. There are four words (8 characters) reserved for each password. The relative position of a password corresponds to the user class number defined in the schema. For user class numbers not defined in the SCHEMA, the four word field is filled with blanks.

Item Read/Write Table



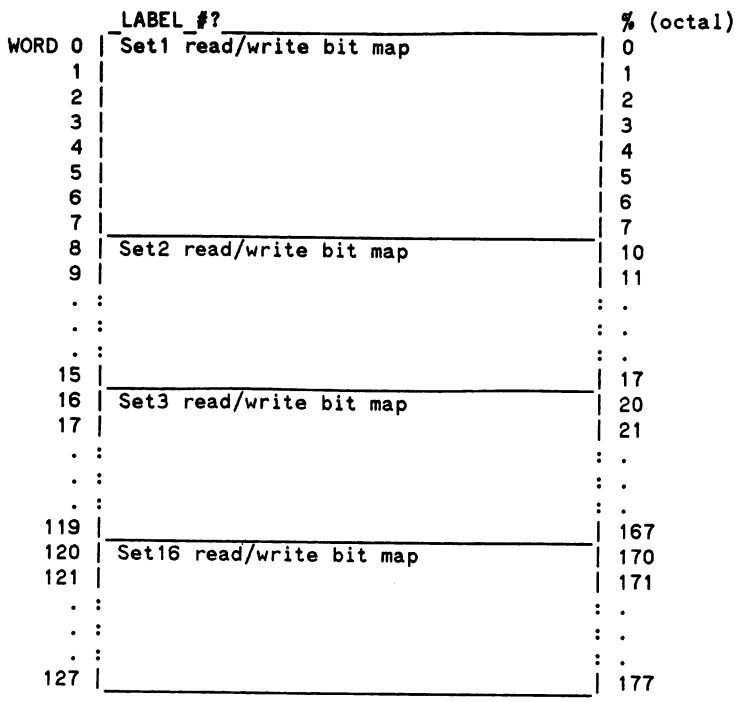
The Item Read/Write Table begins at user label 3. There are eight words for each Item Read/Write bit map. For databases with more than 16 items, the read/write table continues in the next user label. The specific format of this table is explained after the Set Read/Write Table since it is defined the same way. The number of user labels occupied by the Item Read/Write Table depends on the number of data items defined in the schema and can be obtained by rounding up (ceiling) the result of:

Num-of-labels = [(Num-of-items)*8]/128

Since there can only be a maximum of 1023 data items in the schema, the maximum size for this table in user labels would be:

Max-size = [(1023)*8]/128 = 63.93 => 64 labels.

Set Read/Write Table



The Set Read/Write Table begins at a user label boundary following the Item Read/Write Table. There are eight words for each Set Read/Write bit map. For databases with more than 16 data sets, the read/write table continues in the next user label. The specific format of this table is shown in the next page.

The number of user labels occupied by the Set Read/Write Table depends on the number of data sets defined in the schema, and is obtained by rounding up (ceiling) the result of:

Num-of-labels = [(Num-of-sets)*8]/128

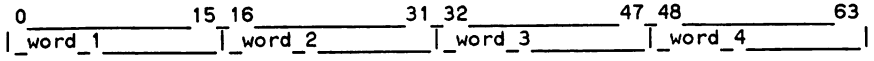
Since there can only be a maximum of 199 data sets defined in the schema the maximum size for this table in user labels is:

Max-size = [(199)*8]/128 = 12.44 => 13 labels

Format for the Read/Write bit maps:

The Read/Write Table has an entry for each item/set in the database. Each entry is 8 words long and up to 16 items/sets per record (user label). Within each 8 words, the first 4 words are the flags for the user classes which have read access to the item/set. The second 4 words are the flags for the user classes which have write access to the item/set. The detailed format for an eight word field is shown below.

A. Four words for read access:

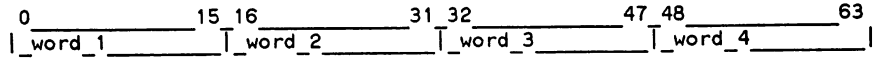


4 words represent 64 bits. Bit n represents read access for user class n to the item/set. If bit n is set to 1 then user class n has read access to the item/set. For example, if the word settings are:

word 1	word 2	word 3	word 4
%000016	%020000	%000410	%001300

This means that user classes 12, 13, 14, 18, 39, 44, 54, 56 and 57 have read access to the item/set. If no read/write security is defined at all for the item/set, then all of the read security bits are set to 1 by default.

B. Four words for write access:



Write access flags have the same format as the read access flags. Bit n represents write access for user class n to the item/set. If bit n is set to 1, then user class n has write access to the item/set For example, if the word settings are:

word 1	word 2	word 3	word 4
%000010	%020000	%000000	%001100

This means that the user classes 12, 18, 54 and 57 have write access to the item/set. If no read/write security is defined at all for the item/set, then all of the write security bits are set to 0 by default.

Database Global Information (Record 0)

	RECORD #0		%
word 0	ROOT'DBSTATUS		0
1	ROOT'DBNAME		1
.			.
4			4
5	ROOT'TRLRLGTH	(trailer area length)	5
6	ROOT'BUFFLGTH	(buffer length)	6
7	ROOT'LGTH	(rootfile length)	7
8			10
9	ROOT'ITEMCT	(number of items)	11
10	ROOT'SETCT	(number of data sets)	12
11	ROOT'ITEMPTR	(record # of item table)	13
12	ROOT'DSETPTR	(record # of set table)	14
13	ROOT'DSCBPTR	(record # of DSCBs)	15
14	ROOT'DEVICEPTR	(record # of device class tbl)	16
15	ROOT'DBGFLAG		17
16	RESERVED	(set to blanks)	20
.			.
19			23
20	NOWOPEN		24
21	MAXOPEN		25
22	RR'RESTART'CALENDAR		26
23	RR'RESTART'TIMESTAMP		27
24			30
25	RR'RESTART'FNAME		31
.			.
28			34
29	RR'RESTART'FGROUP		35
.			.
32			40
33	RR'RESTART'FACCT		41
.			.
36			44
37	RESERVED	(for future use)	45
.		(set to binary 0s)	.
.			.
127			177

ROOT'DBSTATUS - TurboIMAGE:
(0:8) 'C' in ASCII.
(8:8) Octal 2 (filler).
IMAGE (NLS version):
(0:8) 'C' in ASCII.
(8:8) Octal 1 (filler).
IMAGE (pre-NLS version):
(0:8) 'B' in ASCII.
(8:8) Octal 1 (filler).

ROOT'DBNAME: DATABASE name left justified (last 2 chars are blank).

ROOT'TRLRLGTH: The DBG trailer length. This is one of many variables DBOPEN uses to determine the DBG size.

ROOT'BUFFLGTH: The largest buffer size required to accommodate the database blocks.

ROOT'LGTH (+): The size of the root file. This variable was changed from a single word to a double word.

ROOT'ITEMCT: The number of items defined in the database.

ROOT'SETCT: The number of data sets defined in the database.

ROOT'ITEMPTR (+): The Item Table Pointer. It consists of the record number where the Item Table resides.

ROOT'DSETPTR (+): The Data Set Table Pointer. It consists of the record number where the Data Set Table resides.

ROOT'DSCBPTR (+): The DSCB (Data Set Control Blocks) Pointer. It consists of the record number where the DSCB resides.

ROOT'DEVICEPTR (+): The Device Table Pointer. It consists of the record number where the Device Table resides.

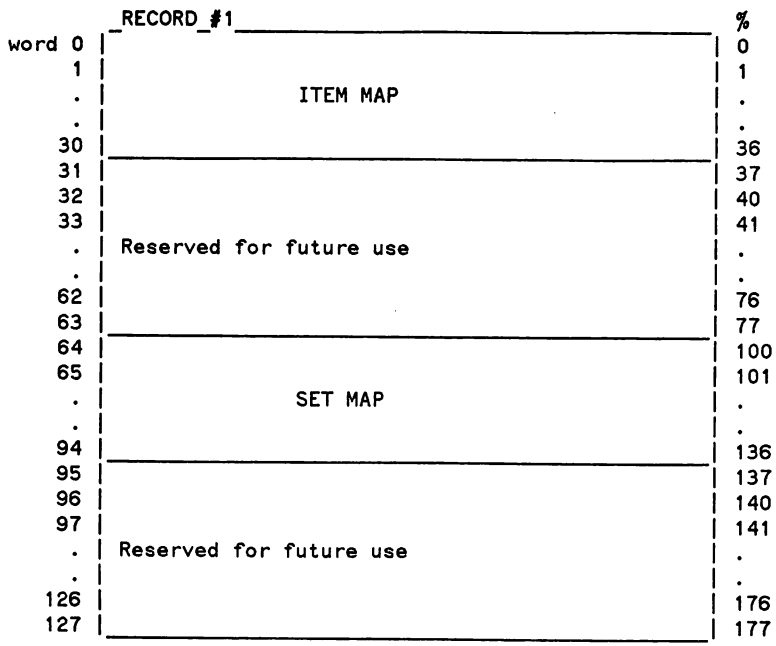
NOWOPEN: Number of data sets opened. This field is NOT used in IMAGE B and TurboIMAGE.

MAXOPEN: Maximum number of data sets that can be opened. This field is NOT used in IMAGE B and TurboIMAGE.

ROOT'DBGFLAG (+): 1: Information can fit in the DBG.
0: Information cannot fit in the DBG.

RR'RESTART'FNAME (+): Restart file name for DBRECOV stop/restart.

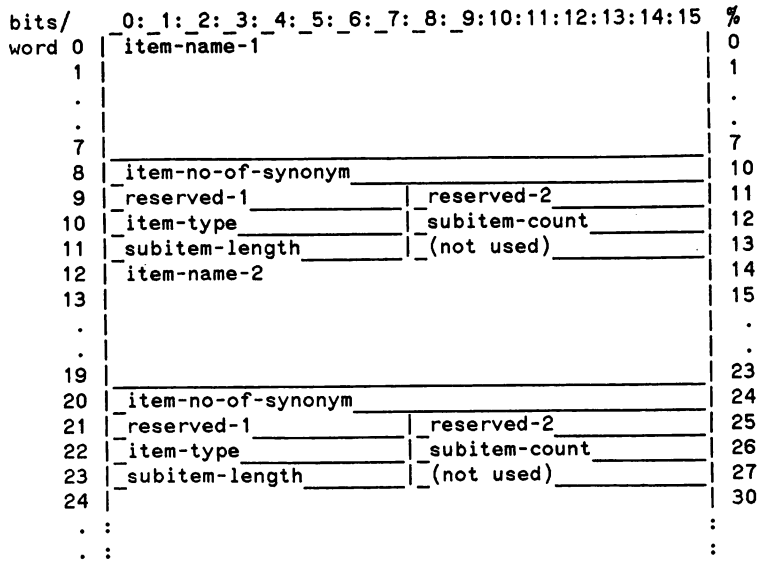
Item and Set Maps (+)



The Item Map occupies words 0 to 30. The Set Map occupies words 64 to 94. These two maps are new in TurboIMAGE. The Item Map is used for searching an item in the Item Table and the Set Map is used for searching a data set in the Set Table.

An item name (or a data set name) is hashed, through the internal hash function, to a double-word value. The final hash value is the modulo for the double-word value by 31 (double-word value MOD 31). All items (or sets) which have the same hash value are chained together. There are total of 31 chains in the Item Table (or Set Table). The Item Map serves as the chain head for each chain in the Item Table and the Set Map serves as the chain head for each chain in the Set Table.

Item Table



The Item Table starts in record 2. Each entry is 12 words long and the length of the table depends on the number of data items defined in the schema. The relative position of an item definition depends on its relative position in the schema.

`item-name:` A data item name, left-justified with trailing blanks.

`item-no-of-synonym (+):` The number of the item whose name has the same hashed result as this one (this is utilized for quick item name searches). This has been changed from a byte to a word.

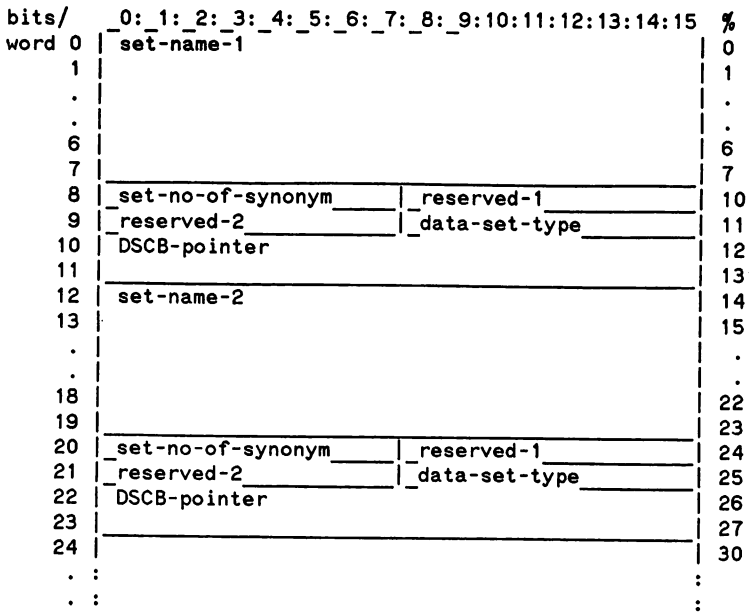
`item-type:` One of the following: I, J, K, R, X, U, Z, or P

```
                item-type
                |
      VALUES, 20J2;
              | |subitem-length
              |subitem-count
```

The maximum size for this table is 12*1023 = 12276 words

`Note:` The reserved-1 and reserved-2 fields are the old level numbers for read and write security. Now, these values are always zero.

Set Table



The Set Table follows the Item Table. It starts at a record boundry. Each entry is 12 words long. The length of the table depends on the number of data sets defined in the schema. The relative position of a set definition depends on its relative position in the schema.

`set-name:` A data set name, left-justified and with trailing blanks.

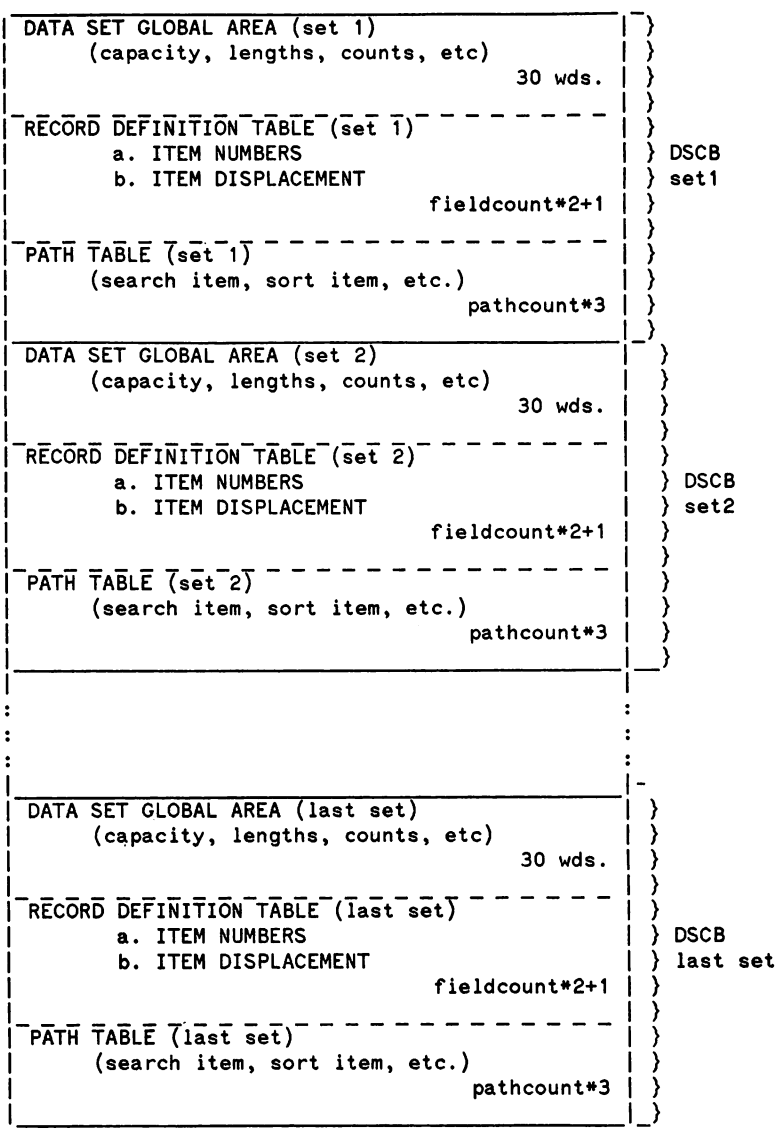
`set-no-of-synonym:` The number of a data set whose name has the same hashed result as this one (this is utilized for quick set name searches).

`data-set-type:` One of the following: A, M or D.

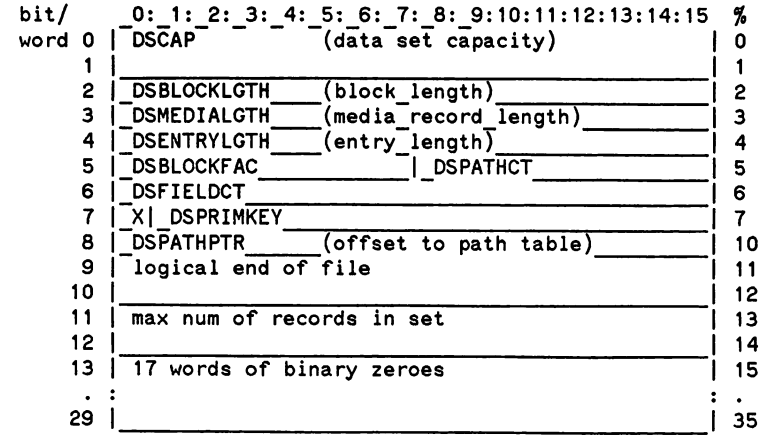
`DSCB-pointer (+):` A pointer to the Data Set Control Block. It has been changed from one word to a double word. The pointer is a word offset from record 0. The DSCB is described on the following pages.

The maximum size for this table is 12*199 = 2388 words.

`Note:` The reserved-1 and reserved-2 fields are the old level numbers for the read and write access respectively. Since this concept no longer applies, the values are set to zero.



The DSCBs follow the Set Table in the Root File. There is one DSCB for each data set defined. The function of the DSCB is to define each data set within the data base.



DSCAP - data set capacity as reported by DBSCHEMA.

DSBLOCKLGTH - data set block length including the bit map overhead.

DSMEDIALGTH - data set media record length (remember that this length includes the pointer overhead)

DSENTRYLGTH - data set entry length.

DSBLOCKFAC - data set blocking factor.

DSPATHCT - data set path count. This is the number of paths that are specified for the data set.

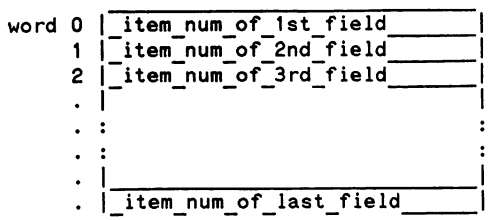
DSFIELDCT(+)- data set field count. This is the number of fields (items) specified for the data set. It has been changed from a byte to a word.

X-DSKEYTYPE - data set key type. If DSKEYTYPE = TRUE then the key is hashed.

DSPRIMKEY - data set primary path or key. For master data sets, this is the field number of the search item. For detail data sets, this is the path number of the primary path.

DSPATHPTR - data set path table pointer. This is a Word offset (relative to its own DSCB) to the data set path table which contains an entry for each path defined. It points to 0th entry in the path table, so to get to the first entry the pointer should be incremented by the length of the entry (which currently is 3 words).

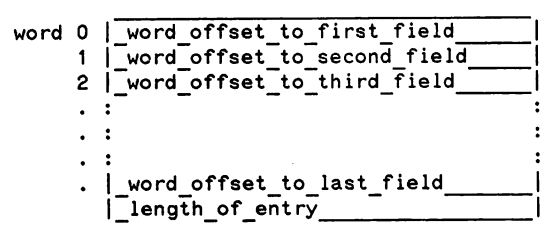
Data Set Control Blocks (Item Numbers)



The Item Numbers Table is part of the Record Definition Table in the DSCB. It follows the Global Area of the DSCB. The size of this table (in words) is equal to the number of items in the data set. This table has been changed from a byte array to a word array to support larger item numbers, for example, 300.

The first field is the first item defined in the data set. The last field is the last item defined in the data set.

Data Set Control Blocks (Item Displacement)

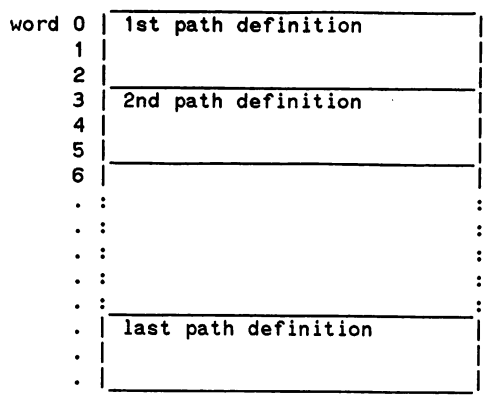


The Item Displacement is also part of the Record Definition Table in the DSCB. It immediately follows the Item Numbers Table.

The word offset points to the starting location of the field within the media record. Remember that the media record includes the pointer overhead so this offset varies for master and detail data sets. If a master data set has only one path, the word offset for the first field is 11, since there are 11 words of overhead (5 words for the synonym chain pointers and 6 words for the data set chain head). For a detail data set with one path, the overhead is only 4 words.

The 'length-of-entry' is the same as the media record length.

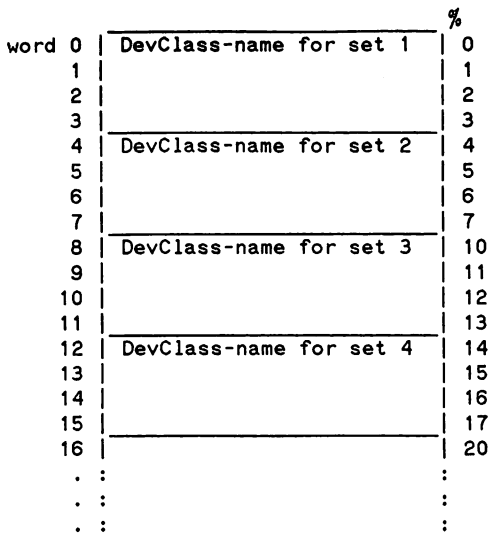
Data Set Control Blocks (Path Table)



The Path Table follows the Record Definition Table in the DSCB. There are 3 words (6 bytes) for each path definition. The Path Table for master data sets has a different layout from the Path Table for detail data sets.

```
Master sets:
        Byte  Description
        1-2 : Item number of the search item in the related detail set.
        3-4 : Item number of the sort item in the related detail set.
          5 : Set number of the related detail data set.
          6 : Path number of the corresponding path in the related
              detail data set.

Detail sets:
        Byte  Description
        1-2 : Field number of the search item.
        3-4 : Field number of the sort item.
          5 : Set number of the related master data set.
          6 : Path number of the corresponding path in the related
              master data set.
```

Device Class Table (+)



The Device Class Table follows the DSCBs. It begins at a record boundry. There is an entry reserved for each data set defined in the schema. Each entry is 4 words long. It contains the device class name which can be optionally specified for the data set in the schema. For data sets without device class names, the entries will be filled with blanks.

This table is created by DBSCHEMA. DBUTIL uses this information to create the data sets. A data set without a device class defined will be created with "DISC" as the default. Once the database is created, the information in this table will NOT be used again. The new DBUTIL command "MOVE" moves a data set from one device to another. However, it does not update the Device Class Table.

The length of the table depends on the number of data sets defined in the schema.

The maximum size for this table is 4*199 = 796 words.

Data Set Structure

A data set is an MPE file, with a file code of -401, created by DBUTIL. It consists of a user label for the data set information, and many records for the user data. The record size by default is 512 words, fixed, binary format with a file system blocking factor of 1. A record is a TurboIMAGE block. The block size can be defined in the schema to be different from the default by using the TurboIMAGE blocking factor and BLOCKMAX options. A different blocking factor can be specified for each data set. Following is an example of defining a blocking factor:

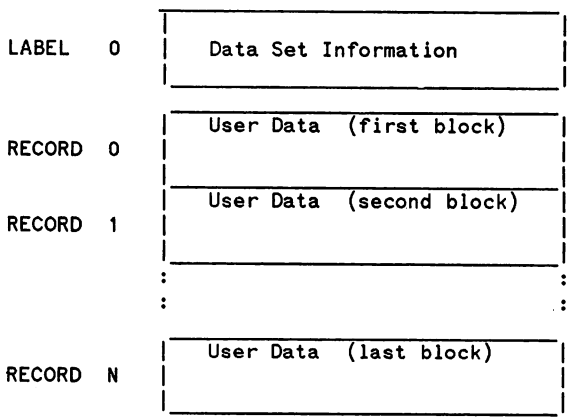
```
NAME:       SETA, Detail
ENTRY:      item1,
            item2,
            item3,
            itemlast;

CAPACITY:   511(22);
```

In the above example, the data set SETA has a capacity of 511 and a blocking factor of 22, i.e. 22 entries in a TurboIMAGE block. The blocking factor is used to define the block size. If the block size is larger than the default, the BLOCKMAX must be specified before the data set:

```
$CONTROL BLOCKMAX=nnnn
```

where "nnnn" is the maximum block size for the data sets that follow this statement.

Following is the general format of a data set:



Data Set Information (Label 0)

The Data Set User Label consists of 6 words of information. The information for a master data set is different from a detail data set.

Master:

word 0-1: The record name* of the last entry allowed in the set, i.e. the capacity of the data set.

2-3: The number of free records in the data set. This number is incremented when a record is deleted and decremented when a record is added. To determine the current number of entries in the data set, subtract Word 2-3 (free records) from the data set capacity.

4-5: Not used.

Detail:

Word 0-1: High Water Mark. This is the record name of the highest record that has been used in the detail set. It is updated by DBPUT only. For example, a detail set with 75 entries and a capacity of 100 has a High Water Mark pointing to the 75th entry. The High Water Mark stays the same even after entries are deleted.

2-3: The number of free records in the data set; this is the same as the master data set.

4-5: The delete chain head for the detail set. It points to the record most recently deleted. It has a value of zero if no records have been deleted from the data set.

* A record name is a double word that consists of a block number (i.e. file system record number) and the slot number in the block. The first 3 bytes is the block number and the last byte is the slot number. A slot is an area in the block allocated for a media record. The first media record in the block is slot number 1. The last media record is slot number N where N is the blocking factor. The first block in the data set is block number 0.

Data Set Blocks

The format of a TurboIMAGE block is the same as described in the IMAGE Reference Manual with the exception of the chain count in the master entries. The chain count has been changed from a single to a double word. The largest number that can be stored in a single word is 65K, thus 65K was the limitation for IMAGE. With a double word, TurboIMAGE does not have this limitation. However, this DOES NOT mean that all entries in the set should be linked in one chain. It would have a negative impact on the performance of chain reads. In fact, it is strongly recommended that all chains be kept short.

Biography

Doris Chen is the R&D project manager for TurboIMAGE, IMAGE, and DBchange. She has been with Hewlett-Packard since 1979.