

# DBGeneral 7.4 Install Instructions

DBGeneral Manual: After installing, please find the DBGeneral Manual in PDF format as DBGPDF.pub.bradmark. To view the file, transfer using WRQ binary to a filename on the PC with an extension of '.pdf'. These release notes are also available as DBG74PDF.pub.bradmark. You must have the Adobe® Acrobat® Reader™ to view the file (available from [www.adobe.com](http://www.adobe.com) free of charge.)

You must be running MPE/iX 6.5 or later to install this version of DBGeneral. You can install DBGENERAL while users are on the system; however, no one should be running DBGENERAL during the install.

DBGeneral installs into the PUB group of the BRADMARK Account, which is created with PM. Your DBGeneral password may change, please verify the new password is correct so that batch jobs, etc. will not fail with a bad password. Please see updated features and functionality in the DBGeneral 7.4 Release Notes document.

The PUBNEW install option allows this version to run with another version of DBGeneral in PUB. On machines where NETBASE is installed, please be sure the NBDIR contains a NONB entry for @.@.bradmark. Where SUPERDEX is installed on the machine, please make an entry @.superdex.sys.

```
:run NBDIR.PUB.NETBASE
D>PROGRAM @.@.BRADMARK;NONB
D>PROGRAM @.SUPERDEX.SYS;NONB
```

## Installation Steps:

- A. Restore the files into the BRADMARK account

```
:HELLO MANAGER.SYS
:FILE T;DEV=TAPE
:RESTORE *T; @.@.BRADMARK;CREATE; show
```

Note: For PUBNEW install, please use

```
:RESTORE *T; @.pub.BRADMARK; GROUP=PUBNEW; CREATE; show
```

- B. Reply to the tape request, if needed

```
:REPLY pin, dat ldev number
```

- C. When this restore has completed, please type

```
:XEQ      SETUP.PUB.BRADMARK
:STREAM   JSETUP.PUB.BRADMARK
```

Note: For PUBNEW install, please use

```
:XEQ      SETUPN.PUBNEW.BRADMARK
:STREAM   JSETUPN.PUBNEW.BRADMARK
```

- D. Please try running DBGeneral to verify the installation, by typing DBGENRL.PUB.BRADMARK. The version number will be 7.4.##.

```
:DBGENRL.PUB.BRADMARK
```

Note: For PUBNEW install, please use

```
:DBGENRL.PUBNEW.BRADMARK
```





[www.bradmark.com](http://www.bradmark.com)  
[support@bradmark.com](mailto:support@bradmark.com)  
24-hour Technical Support  
713-621-2808, 800-621-2808

## DBGeneral 7.4 Release Notes

Organization of the release bulletin highlighting the changes to DBGeneral for IMAGE:

### Capacity Change New or Enhanced Features Overview

- Options At 'New Capacity' Prompt
- Options At 'New Jumbo Capacity' Prompt
- New Pre-Processing Speeds Character Key Type Master Capacity Changes
- Jumbo Master Dataset Features
- Master Dynamic Dataset Expansion (MDX)
- Ventilation Yields Better DBPUT Performance
- Faster Master Capacity Change Via Intelligent Secondary Placement Technology
- Option 3.2 Auto Capacity Management Supports JUMBO Datasets



**DBGENALT** enhanced for JUMBO and Dynamic datasets, B-trees, and compatibility with ONLINE Backup

### Enhanced Diagnostics and Repair

- Faster Path Level Diagnostics
- User-Selectable Path Analysis
- Display Of Key Values For Uncovered Errors
- User-Selectable Sort Field Option In Path Repair
- Diagnostic Integration Providing Selective Key Value Repair

### Support for Changes in IMAGE

- Support for IMAGE Expansions: Master Paths, Items, Sets, Pointer Format

### Dialog Changes in DBGeneral

- Master Capacity Change (option 3.3) now prompts 'Is this capacity correct' in Batch

### DBGeneral NEW, Enhanced or Repaired Features Overview

- Improved Various Read Only Options to Not Changed the MODIFY Date of Datasets
- Option 1.4 Dynamic Master Dataset Support
- Option 1.6 Supports Lock Descriptors for Very Large Keys
- Option 2.1 and Option 2.2 Jumbo Master Support, Dynamic Master Set Capacities
- Option 2.4 False Error When Delete Chain Length Equals One
- Option 2.4 Single Path Analysis, 50% Faster
- Option 2.5/4/5 Complete Path Rebuild, Supports User-Selectable Sort Field
- Option 2.6 Opens Database Exclusive by Default
- Option 2.6 Computing Environment Information and \$stdlist enhancements
- Option 3.3 IMAGE Root File Version Update
- Option 3.5 Detail Dataset 80 Gig Support
- Option 3.5 Compatibility with ONLINE BACKUP jew DBG35COPY
- Option 3.6 Choice Of Path During Reorganize (Repack) Detail Dataset
- Option 3.6 Reorganize (Repack) Detail Dataset Facilitates Minor Repairs
- Option 4.4 Erase Dataset Manages MDX and DDX
- Option 4.6 Copy of Base Open for Read Access Now Avoids Subsequent DBOPEN errors
- Option 4.7 permits direct specification of blocking factor
- Option 5.# Specification of Database Changes Permits Placement Before or After an Entity
- Option 8.1 DB Key Change Supports Critical Item Update (CIUpdate)
- Improved Third Party Indexing processing for OMNIDEX

# Master Capacity Change New or Enhanced Features

## Options at the 'New Capacity' Prompt

During DBGeneral's capacity change, a consistent dialog is provided whether you are doing a master or detail capacity change. The user response to the prompt 'New Capacity' is key to controlling the type and size of the new dataset created.

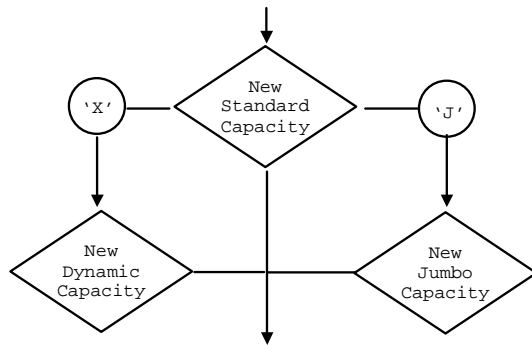
```

New capacities may be specified in any of the following formats :

Absolute capacity      : number          (e.g. "5501")
Relative change       : +/- change      (e.g. "+500")
Percent change        : +/- change %    (e.g. "-25%")
Percent of free space : number F        (e.g. "30F")
DDX Dynamic Dataset eXpansion parms (e.g. "X")
Jumbo data set parameters (e.g. "J")
    
```

7.4 Option 3.3 Jumbo Master Capacity Change 'new capacity options'

The dialog describes the current dataset type and characteristics, and then prompts for a new capacity. Converting a dataset from standard to Jumbo, from Standard to MDX (for masters) or DDX (for details), and retaining the Jumbo or MDX / DDX dataset type is accomplished by typing 'J' or 'X' respectively at the 'new capacity' prompt. Once it is established that the dataset type will remain the same or be converted, the capacity dialog for that type of dataset proceeds.



From Dataset	To Dataset	Permitted
Standard	Jumbo	OK
Standard	DDX / MDX	OK
DDX / MDX	Standard	OK
Jumbo	Standard	OK
Jumbo	DDX / MDX	No
DDX / MDX	Jumbo	No

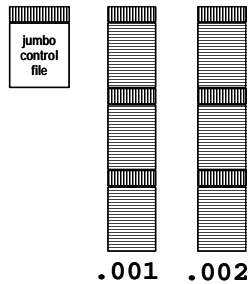
## Options at the 'New Jumbo Capacity' Prompt

DBGeneral's Jumbo capacity change permits conversion to Jumbo for small datasets to permit testing and allowing explicit control of chunk sizes for larger datasets. The 'New Jumbo Capacity' prompt has two parameters: Jumbo capacity and entries per chunk. Therefore a new jumbo capacity specification of '5000000,2500000' will produce a jumbo dataset with two chunks each containing 2,500,000 entries. For small database testing of Jumbo, a new jumbo capacity specification of '500,250' will produce a Jumbo dataset with two chunks each containing 250 entries. This feature takes DBSCHEMA's '\$CONTROL FORCEJUMBO' command, which allows the creation of small Jumbo datasets, one step further by allowing explicit control of chunk size.

## New 'Pre-Processing' Speeds Character Key Type Master Capacity Changes

With an eye toward performance, DBGeneral now automatically invokes the pre-processing option for master capacity change on character keys when the number of entries exceeds 250,000. Previously this option was invoked via a non-zero value for the JCW DBG33SORT. Subsequently, it has been determined that there is a performance benefit in most capacity changes using the pre-processing and it has therefore been made the default.

## Jumbo Master Dataset support



Jumbo master datasets provide capacity for the largest of masters via the HFS file domain by utilizing multiple chunk files (suffixed '.001', '.002', etc.) beneath a single chunk header file. This presents challenging performance problems due to the volume of data. DBGeneral solves these issues through the use of Bradmark's storage management technology, designed specifically to meet the needs of user's with large databases that require a high-performing, high-capacity tool to manage their database environment.

Sample Dialog of Standard dataset conversion to Jumbo Master dataset Capacity is shown below with comments.

```

DATASET: EXTENDED-M

Search Item Information:
  Name: CLAIM-NO
  Type: X
  Loc: 1
  Length: 4

New capacities may be specified in any of the following formats :
  DDX Dynamic Dataset eXpansion parms      (e.g. "X")
  Jumbo data set parameters                (e.g. "J")

Current set capacity      : 21800003   (73.5% full)
Current set entries      : 16020804
Current blocking factor  : 11
Current block size      : 1024 words

Current file size       : 15854560 sectors
Enter new capacity      : J

Enter new jumbo capacity : 31000000
Proposed capacity       : 31000000   (51.7% full)

The following options are available :

  1 .. Round capacity up to nearest prime   : 31000003
  2 .. Round capacity down to nearest prime : 30999989
  3 .. Fine tune capacity thru Capacity Sampling
      (Note: This feature is not available for JUMBO Sets)
  4 .. Use capacity specified
Enter option required (1:4) :1

Proposed jumbo capacity : 31000003   (51.7% full)
Proposed file size      : (Jumbo Dataset)

      Chunk No.   Entries   Sectors
      -----   -
      Chunk Hdr.   N/A      32 (New)
      1           23067968  16776720 (Old/Exp)
      2           7932035   5768768 (New)
      -----   -
      Totals:     31000003  22545520
      Change:     +9200000  +6690960

Is the new jumbo capacity of 31000003 correct ? (Y/N) : Y

Capacity change in progress ---

Database access disabled.

```

“ J ” indicates new dataset will be  
**JUMBO**

The proposed file description gives full information about the changes to the dataset.

In this example, Chunk No. 1 is the old normal Master dataset renamed into the HFS space and is expanded to hold 23,067,968 entries requiring 16,776,720 sectors, hence the status (Old/Exp).

Chunk No. 2 is (New) and will hold 7,932,035 entries requiring 5,768,768 sectors.

Totals are shown as 31,000,003 entries requiring 22,545,520 sectors. This is a change of +9,200,000 entries and +6,690,960 sectors.

7.4 Option 3.3 Jumbo Master Capacity Change 'Setup Dialog'

```

NM Cap change 7.4.00 (02/24/00) (c) Copyright 1997, Bradmark.

All preparations done, extracting entries...
TUE, SEP 28, 1999, 11:54 PM

 10% complete. Cumul: 0:01:37 Remain: 0:14:36 Rate: 16457 Ent/sec.
 20% complete. Cumul: 0:03:14 Remain: 0:12:18 Rate: 17178 Ent/sec.
/snip
 90% complete. Cumul: 0:20:34 Remain: 0:02:19 Rate: 11659 Ent/sec.
100% complete. Cumul: 0:23:52 Remain: 0:00:00 Rate: 11183 Ent/sec.

Number of entries extracted : 16020804
Wall time: 0:23:52 CPU time: 0:09:18.885

Invoking sort for entries.
Number of records sorted : 16020804
Wall time: 0:34:11 CPU time: 0:22:09.970

```

Entries are pre-processed to minimize disc IO and maximize data throughput. This pre-processing is enabled for non-numeric key types when the number of entries exceeds 250,000.

7.4 Option 3.3 Jumbo Master Capacity Change 'preprocessing entries'

```

Processing sorted entries...
WED, SEP 29, 1999, 12:52 AM

 10% complete. Cumul: 0:04:21 Remain: 0:39:10 Rate: 6133 Ent/sec.
 20% complete. Cumul: 0:10:58 Remain: 0:43:53 Rate: 4868 Ent/sec.
/snip
 90% complete. Cumul: 0:57:36 Remain: 0:06:24 Rate: 4171 Ent/sec.
100% complete. Cumul: 1:02:25 Remain: 0:00:00 Rate: 4277 Ent/sec.

Number of primaries processed : 8708167
Wall time: 1:02:32 CPU time: 0:14:03.607

```

Primaries are placed into the new dataset at maximum speed.

The Secondaries are placed into the dataset faster than ever. See 'Faster Master Capacity Change via ...' in these release notes.

7.4 Option 3.3 Jumbo Master Capacity Change 'processing PRIMARIES'

```

Now processing 7312637 secondaries.
WED, SEP 29, 1999, 1:54 AM

Entry level ventilation factor is: 2

 10% complete. Cumul: 0:04:26 Remain: 0:13:53 Rate: 2743 Ent/sec.
 20% complete. Cumul: 0:10:48 Remain: 0:11:29 Rate: 2256 Ent/sec.
/snip
 90% complete. Cumul: 0:52:22 Remain: 0:05:49 Rate: 2094 Ent/sec.
100% complete. Cumul: 0:59:45 Remain: 0:00:00 Rate: 2039 Ent/sec.

Number of secondaries processed : 7312637
Wall time: 0:59:45 CPU time: 0:13:24.434

```

Notice the dataset has been ventilated. See 'Master Dataset Ventilation...' in these release notes.

7.4 Option 3.3 Jumbo Master Capacity Change 'processing SECONDARIES'

```

Total number of entries processed: 16020804
Overall statistics: Wall time: 3:00:23 CPU time: 0:58:56.938

Dataset change successfully completed

```

Overall statistics are presented, including all setup and processing.

7.4 Option 3.3 Jumbo Master Capacity Change 'Overall statistics'

Converting from JUMBO master to normal master is accomplished via a capacity change where 'J' is not specified at the capacity prompt. This will initiate a warning message to prevent inadvertent conversion.

```
WARNING ! Specifying the new capacity without using the 'J'
           option will convert the jumbo fileset back to a 'normal' set
```

7.4 Option 3.3 Jumbo Master Capacity Change 'Warning ! ...conversion back to normal set'

In the next example, a Jumbo Master's super-size is no longer needed due to archiving and the master is converted to a standard master dataset. DBGeneral displays the file's description showing the Jumbo chunks currently allocated. Capacity is specified as '30F', which indicates thirty percent free space should remain in the new dataset. This results in a nine million plus sector savings during the forty-five minute elapsed time processing the five-million plus entries of numeric key type.

```
NOTE!! Dataset is currently configured as a Jumbo Data set

Current set capacity      : 15041239   (38.3% full)
Current set entries      : 5768219
Current blocking factor  : 5
Current block size       : 896 words
Current file size        : (Jumbo Dataset)

      Chunk No.   Entries   Sectors
      -----   -
      Chunk Hdr.   N/A      32
           1     11983360  16776720
           2     3057879   4281040
      -----   -
      Totals:     15041239  21057792

Enter new capacity      : 30F

WARNING ! Specifying the new capacity without using the 'J'
           option will convert the jumbo fileset back to a
           'normal' set.

Proposed capacity       : 8240329   (70.0% full)
Proposed file size      : 11536464 sectors
Change in file size     : -9521328 sectors

Is the new capacity of 8240329 correct ? (Y/N) : Y

Capacity change in progress ---
Database access disabled.

NM Cap change 7.4.00   (02/24/00) (c) Copyright 1997, Bradmark.

/SNIP
Total number of entries processed: 5768219
Overall statistics: Wall time: 0:45:48 CPU time: 0:31:30.046

Dataset change successfully completed

DATASET:
```

Not specifying 'J' causes the conversion by directing DBGeneral to create a normal dataset.

Jumbo conversion warning message is displayed.

7.4 Option 3.3 Jumbo Master Capacity Change 'example'

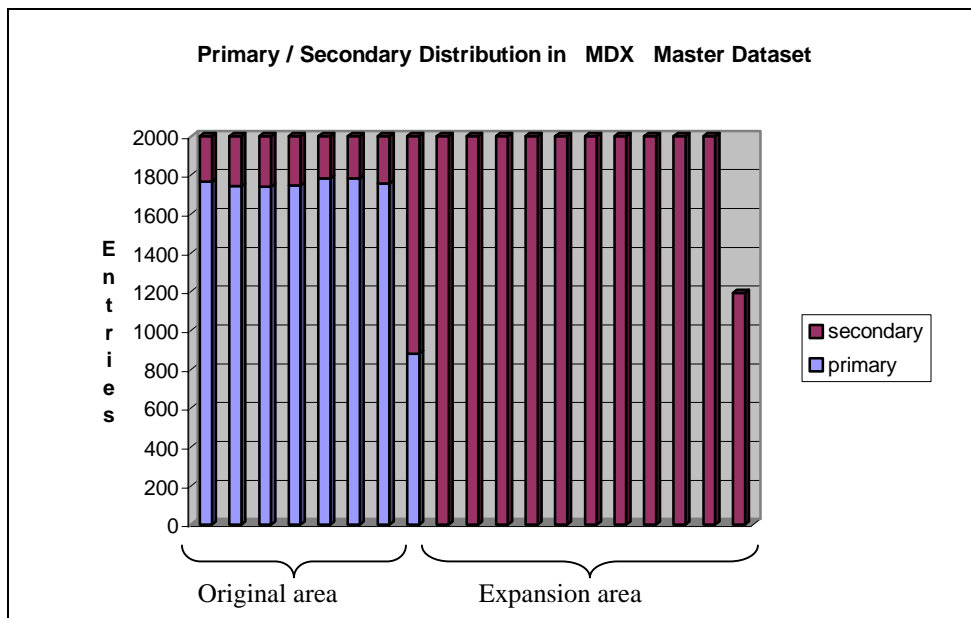
## Master Dynamic Dataset eXpansion (MDX)

Hewlett Packard's implementation of Master Dynamic Dataset eXpansion (MDX) support provides increased resiliency for IMAGE databases by deferring master dataset full conditions. MDX also offers performance improvements on datasets experiencing excessive clustering by halting of the serial scan for secondary placement via 'quick search' (see Hewlett Packard's MPE/iX 6.0 communicator for additional information.) This feature requires IMAGE version C.07.17 and it is recommended that you update to the most current version of IMAGE possible (contact Hewlett Packard). BRADMARK recommends at least IMAGE version C.07.25.

The implementation of Master Dynamic Dataset expansion (MDX) is similar to Dynamic Dataset expansion for Details (DDX). You may generate a new database with MDX from a schema or use a third party database management tool such as DBGeneral to convert an existing master dataset to MDX. DBGeneral's master capacity change now contains features for implementing MDX. By selecting MDX during a master dataset capacity change, the user is directed by DBGeneral to a dialog including maximum capacity, increment capacity and hash capacity which define the parameters of the new dataset.

MDX is also significantly different from DDX. First, the triggering of an expansion is different. A DDX expansion is triggered when the currently allocated capacity of the dataset is full, whereas the initial MDX expansion is triggered when a location for a secondary cannot be found in a reasonable search (IMAGE's 'Quick Search'). Once the MDX dataset has been expanded, the dataset has two distinctly different areas: the original area and expansion area. The original area remains hashed while the expansion area is managed with a delete chain and a high water mark, similar to a DDX dataset.

While DDX may be configured to allow the dataset to grow indefinitely and the expansion area functions identically, a MDX dataset's expansion area is completely different from the original area. Since ONLY secondaries can occupy the expansion area, the dataset will hash all keys as secondaries once the original area is full and may result in poor performance.



Above a MDX dataset's entry count by primary/secondary type is shown the data taken from DBGeneral's option 2.1. Each bar represents one twentieth of the dataset's capacity. The original area is mostly primaries. When this area fills all subsequent entries become secondaries. Notice the expansion area is ALL secondaries. This dataset is an extreme intended for demonstration purposes only.



Below is a sample dialog of a MDX dataset conversion. The 'X' at the first capacity prompt indicates the resulting dataset is intended to be eXpandable. This dialog is similar to the dialog for DDX datasets, except of course, this is a master dataset.

```

DATASET: CONTROL-1

Search Item Information:
  Name: CONTROL-1
  Type: J
  Loc: 1
  Length: 2

New capacities may be specified in any of the following formats:
  DDX Dynamic Dataset eXpansion parms (e.g. "X")
  Jumbo data set parameters (e.g. "J")

Current set capacity      : 200003 (97.4% full)
Current set entries      : 194831
Current blocking factor  : 156
Current block size      : 2048 words
Current file size       : 20544 sectors

Enter new capacity      : X

Enter new DDX initial capacity : 200000

Proposed capacity      : 200000 (97.4% full)
The following options are available :
  1 .. Round capacity up to nearest prime : 200003
  2 .. Round capacity down to nearest prime : 199999
  3 .. Fine tune capacity thru Capacity Sampling
  4 .. Use capacity specified
Enter option required (1:4) : 1
Proposed initial capacity: 200003 (97.4% full)

Proposed file size      : 20544 sectors
Change in file size     : 0 sectors
Is the new DDX initial capacity of 200003 correct ? (Y/N) : Y

Enter new DDX maximum capacity : 280000
Capacity rounded up to a multiple of the set's blocking factor.
This conforms to IMAGE's requirement for Dynamic Expanding sets.
Proposed maximum capacity: 280020 (69.6% full)
Proposed file limit     : 1795 records
Change in file limit    : +512 records

Is the new DDX maximum capacity of 280020 correct ? (Y/N) : Y

Enter new DDX increment : 20000
Increment rounded up to a multiple of the set's blocking factor.
Proposed increment: 20124 (9% of initial)

Is the new DDX increment of 20124 correct ? (Y/N) : Y

```

“ X ” specifies the new dataset will be **MDX**

MDX configuration prompts for:  
 a) Initial or HASH capacity  
 b) Maximum capacity  
 c) Increment capacity

DBGeneral displays information regarding disc space usage of the new dataset, disc space change from old dataset to new dataset, etc. to guide the user through the MDX configuration process.

7.4 Option 3.3 MDX Master Capacity Change 'Setup Dialog'

Master dataset capacity changes performed on MDX datasets will display the current MDX parameters.

```

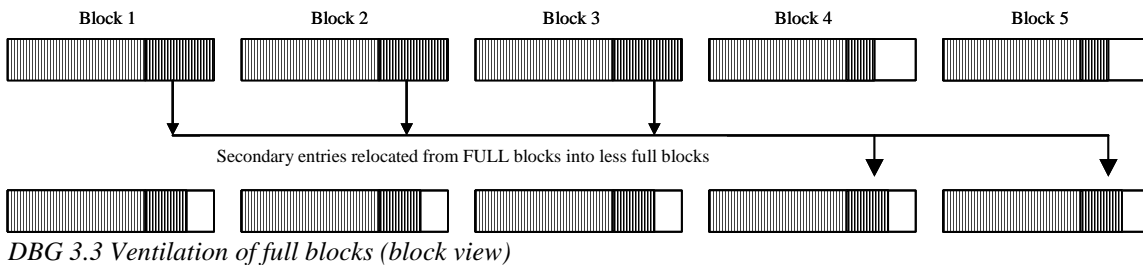
NOTE!!Dataset currently configured for Dynamic Dataset eXpansion
DDX current capacity : 200003 (97.4% full)
DDX maximum capacity : 280020 (69.6% full)
DDX hashing capacity : 200003
DDX increment entries : 20124

```

7.4 Option 3.3 MDX Master Capacity Change 'Dynamic Dataset parameters'

## Ventilation Yields Better DBPUT Performance

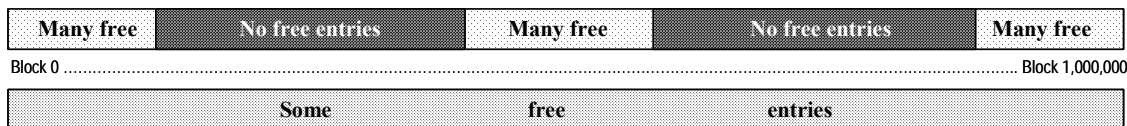
In response to user requests to provide improved master dataset performance, BRADMARK identified datasets with full blocks as a performance bottleneck. In fact, when there are contiguous full blocks of substantial numbers, performance can suffer dramatically. DBPUTs that previously required a fraction of a second can become laborious serial scans searching for available free entries in the master (sometimes requiring minutes, or more.) By dynamically managing secondary placement, DBGeneral's 'Ventilation' is a process whereby master datasets may have significant improvements in performance due to reductions in contiguous full blocks.



The above diagram illustrates the concept of ventilation using a small five-block sample master dataset. Ventilation during secondary processing causes some of the secondary entries in the full blocks to be placed into the less full blocks. Before the ventilation the first three blocks are full and the last two have significant free space. After the ventilation, there are no full blocks and all five blocks have approximately an equivalent number of free entries. Ventilation has successfully eliminated the negative effects of full blocks on performance.

Bradmark's new ventilation technology works in conjunction with DBGeneral's intelligent placement algorithm to minimize disc I/O activity. Together, this new combination of technology improves the performance of master sets when adding new entries as well as retrieving existing entries.

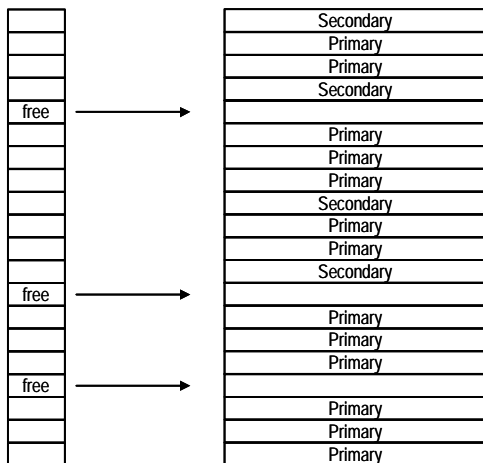
Eliminating contiguous full blocks provides superior performance for DBPUT by increasing the likelihood that a location in the primary's block will have a free entry for the secondary to be placed. DBPUT spends less time scanning IMAGE blocks looking for an available free entry. Datasets that hash poorly and create many contiguous full blocks containing secondaries will perform better after a DBGeneral capacity change with ventilation. While this remedy's benefit will be eroded as IMAGE places more secondaries into the blocks once again filling them, the benefits can be regained by doing another capacity change.



*DBG 3.3 Ventilation of full blocks (dataset view)*

Block free space is like a breath of fresh air to a master dataset and this is where the name 'ventilation' has its origin. Think of a hot, smoky, stale master dataset block where there is no fresh air to breath and no free space. DBGeneral's master capacity change ventilates the block, opening a window on performance.

## Faster Master Capacity Change via Intelligent Secondary Placement



DBGeneral uses its own intelligent secondary placement technology to insure that no time is lost looking for a free entry during secondary processing. Secondary placement is no longer subject to slower run times due to large numbers of full blocks in the locality of the primary. The secondary placement has a pre-qualified 'target' entry into which it places the secondary. The target entry is known to be free, and once utilized is taken off the free entry target list. This method eliminates searching for a free entry to place a secondary via serial processing (which is what IMAGE does). Processing speeds for secondaries reflect this new technology.

Even the most egregious datasets, with large portions of the dataset's blocks filled one hundred percent, exhibit no secondary processing degradation. At the extreme, archive databases' master datasets may be processed by DBGeneral's capacity change such that the dataset is full, using the minimum amount of disk space, without slowing the processing speed of secondary placement. DBGeneral knows where the free entry is and places the secondary at that location. Even if only one free entry remains in the entire dataset, DBGeneral goes directly to that entry. Intelligent Secondary Placement Technology does not waste any I/O bandwidth and is extremely efficient and fast.

## Option 3.2 Auto Capacity Management supports JUMBO MASTER datasets

DBGeneral's Auto Capacity Management has been updated to handle JUMBO master dataset and JUMBO detail datasets. This provides the same auto capacity management features on JUMBO sets as has been previously available on standard datasets. Conversions to/from JUMBO datasets are by default not facilitated. However, setting JCW DBG32JUMBO = 1 will cause option 3.2 to convert master and detail datasets to/from JUMBO as the new capacity dictates. Auto Capacity Management is not available on Dynamic Datasets since the intention is that they manage themselves. Empty datasets may now be included in Auto Capacity Management by setting the JCW DBG32EMPTY = 1.

```

Data Base: BIGDB.OPT32.ATDBG02
Sets:          Type Capacity  Entry  --- Minimum ---  -- Maximum ---
                Count      Level Compress  Level  Expand
ORDER-LINES    DJ 18096352 15967359    0      0  9048176 18277328
**** Previous set requires expansion
**** Capacity will be changed to 18277328
    Capacity change in progress ---
    Database access disabled.

Jumbo Data Set Resize 7.02.03 (12/04/96) Copyright 1996, Bradmark.
Starting jumbo fileset transformation...
Bypassing existing chunk #1 (no chg.)...
Expanding existing chunk #2...
Creating new Chunk Control Header (CCH)...
Chunk Control Header built successfully.
Jumbo fileset transformation complete.

Dataset change successfully completed
Option selected (# / ? / MPE / Menu / Next / End) :
```

*DBG 7.4 Auto Capacity Management on JUMBO datasets*

## DBGENALT Command for DBGeneral version 7.4

DBGENALT is a command file that requires MPEX from VESoft Inc. This command file can be used to manage capacities of databases by using MPEX to control the fileset being managed and DBGeneral to perform the capacity change.

### SYNTAX

`%DBGENALT fileset, percent, password`

*fileset* may be any legal MPEX fileset; however, only datasets in the logon group created by the logon user will be managed.

*percent* ratio of entry count over new capacity. Specifying .7, indicates the set's capacity should be adjusted such that the set is 70% full.

*Password* DBGeneral password

### Sample command

```
:mpex
%DBGENALT @.macsdata.sgaii(dbsetfullness>.7), .7, testa
Changes all datasets in macsdata that are more than 70% (.7) full, to become 70% full.
```

### DBGENALT has been enhanced to support:

DBGENALT now supports JUMBO Masters, JUMBO Details, Dynamic Master Datasets, Dynamic Detail Datasets and Masters with B-trees. Previous versions encountered problems with the MPEX looping constructs such that the same dataset was changed multiple times and others not at all.

By default, DBGENALT will not manage JUMBO or Dynamic datasets; however, you can tell DBGENALT that you want to manage JUMBOs or Dynamic dataset by setting the variables shown below. Note that by default dynamic sets have their maximum capacity set equal to 5 times the new capacity and the increment to one tenth the new capacity. These defaults can be changed directly in the command file.

```
:SETVAR DBGENALT_JUMBO TRUE
:SETVAR DBGENALT_DYNAMIC TRUE
```

DBGENALT has been enhanced to automatically enable the DBG35COPY option when a dataset is being STORED with the ONLINE option at the time of the capacity change.

### DBGENALT Installation

DBGENALT is found in the MPEX group of the BRADMARK account (dbgenalt.mpex.bradmark) and executed by specifying the fully qualified file name. You must copy this file into the cmd22 group of the VESoft account (dbgenalt.cmd22.vesoft) and execute by typing 'DBGENALT'. If you try to use the file as DBGENALT.MPEX.BRADMARK, MPEX will see that there is a file DBGENALT in cmd22.vesoft and use it instead !

## DBGENALT will manage databases in one group and one creator

DBGENALT focuses on managing one or more databases created by the logon user in one group. Previously, DBGENALT would permit multiple accounts, groups and creators by using a recursive algorithm to stream itself for each unique creator/group/account of the fileset specified. This generally worked well; however, there were some issues. At times this spawning effect would produce a confusing web of jobs. Also, multiple jobs may be streamed in the same group by the old DBGENALT's recursive behavior, which is not permitted by DBGeneral and must be avoided with job queues or limits. Overall, most users seem to be using DBGENALT to manage a single database or several databases in one group that all have the same creator.

## DBGENALT Dialog messages

```
%DBGENALT @,.7,testa
Version 2.06 For Standard, JUMBO, B-tree and Dynamic datasets.
Manage Dynamic Sets: FALSE   Manage Jumbo Sets: FALSE
syntax dbgenalt fileset(selection criteria), percent-full, dbgeneral-password
e.g. %DBGENALT @(dbsetfullness>.7),.6,testa ((fuller than 70%, change cap to 60%))

    Not an IMAGE related file::: DBGCMD.DBGENALT.ATDBG(0)
    Not an IMAGE related file::: DBGCMDA.DBGENALT.ATDBG(0)
    Not an IMAGE related file::: DBGVE.DBGENALT.ATDBG(0)
    Not an IMAGE related file::: DBGVEE.DBGENALT.ATDBG(0)
    JUMBO flag off, not processed::: DBX01.DBGENALT.ATDBG
    Creator different than current logon::: DBX02.DBGENALT.ATDBG (MANAGER:::MGR)
    Dynamic flag off, not processed::: DBX03.DBGENALT.ATDBG
    Processing Standard dataset::: DBX04.DBGENALT.ATDBG
    Processing Standard dataset::: DBX05.DBGENALT.ATDBG
    Processing Standard dataset::: DBX06.DBGENALT.ATDBG
```

**Not an IMAGE related file:::**

Indicates that currently processed file in the file set is not an IMAGE dataset. The file name is displayed along with the file code in parenthesis.

**JUMBO flag off, not processed:::**

Indicates that the currently processed file in the file set is a JUMBO dataset (Master or Detail) and the JUMBO management option is not enabled. To cause DBGENALT to manage JUMBO datasets, please 'SETVAR DBGENALT\_JUMBO TRUE'.

**Creator different than current logon:::**

Indicates that the currently processed file in the file set has a creator different than the current logon. DBGENALT must be run from a logon that is the creator of the database. The file name is displayed along with the file creator and current logon user name (*creator user name ::: logon user name*).

**Dynamic flag off, not processed:::**

Indicates that the currently processed file in the file set is a Dynamic dataset (Master or Detail) and the Dynamic management option is not enabled. To cause DBGENALT to manage Dynamic datasets, please 'SETVAR DBGENALT\_DYNAMIC TRUE'.

**Processing Standard dataset:::**

Indicates that the currently processed file in the file set will be managed.

# Enhanced Diagnostics and Repair

## Faster Path Level Diagnostics

Insuring that a database is structurally sound can be a time consuming and laborious task. One of the most resource intensive types of database diagnostics are detail path chain walking which is done to insure that every entry in the detail is linked correctly to the corresponding master and same key value detail entry. DBGeneral's Native Mode Dataset Diagnostics (option 2.4) has been enhanced to maximize the speed of this task. While utilizing a high-speed chain walk as a foundation for this feature, currently the data has been organized to reduce disk I/O. The performance enhancement has been dramatic and makes option 2.4 the diagnostic of choice for detail structural integrity evaluation.

## User-Selectable Path Analysis

Typically all paths are analyzed to insure complete structural integrity of a detail dataset. However, during a repair sequence or if one path is known to be bad, it is useful to be able to specify the path that should be evaluated. DBGeneral's option 2.4 now offers this feature. The user may specify a single path or '@' to process all paths. When only one path out of five requires integrity checking, this enhancement may produce a five-fold increase in performance.

Path #	Search Key	Related Master
1	SERIAL-NUMBER	SERIAL-NUMBER-A
2	ACCOUNT	ACCOUNT-A
3	LAST-NAME	LAST-NAME-A
4	DATE-LAST	DATE
5	DATE-FIRST	DATE

Enter path number to analyze or '@' for all paths :

7.4 Option 2.4 'Path selection'

## Display of Key Values for Uncovered Structural Flaws

When an ERROR ! is discovered by the path analysis, DBGeneral has been enhanced to display the key values in both HEX and ASCII in addition to the record number of the entry in the dataset. This information makes reading the ERROR ! report more informative and useful in evaluating the nature of the corruption.

```
ERROR ! Orphaned detail entry located at 5573448
Detail Key: >3207924420 <
(Hex): >3332 3037 3932 3434 3230 2020 2020 2020 2020 2020<

ERROR ! Invalid master bkwd chain pointer, master record 43
Master Key: >0500050005000500050005000500050005<
(Hex): >3035 3030 3035 3030 3035 3030 3035 3030 3035 3030 3035 3030 3035 3030 3035<

ERROR ! Orphaned detail entry located at 1
Detail Key: >0100010001000100010001000100010001<
(Hex): >3031 3030 3031 3030 3031 3030 3031 3030 3031 3030 3031 3030 3031 3030 3031<
```

7.4 Option 2.4 'ERROR ! samples'

## User-Selectable Sort Field Option in Path Repair

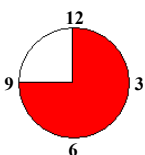
DBGeneral has been enhanced to permit option 2.5 'complete path rebuild' to sequence the chained data in a manner similar to a sort item. This is the equivalent of a sort-item, but is only implemented during the path re-build and subsequent entries added to the dataset are not sorted. Option 2.5 has always enforced sort-items. This new feature is useful when data requires sequencing which has not been implemented as a sort-item.

## Diagnostic Integration Providing Selective Key Value Repair

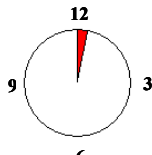


When datasets become corrupt due to a system failure or some other cause, sometimes only a few chains out of thousands or millions may be affected. This damage may go undetected for years, until an archive program visits the old data, or it may be uncovered immediately. When the problem is detected, the database administrator must find the exclusive access (users out of the database) time to make the repairs. Since repairing some extremely large datasets can be time consuming, the administrator must weight the cost of a corrupt database causing an occasional application abort/problem verses the impact of database down time for repairs. If this is a mission critical database, the problem is particularly difficult.

Enter DBGeneral's Native Mode Diagnostic's partnership with Native Mode Path Repair! This enhancement allows these two features to communicate via the creation of data files that indicate the corrupted key values for a particular path. These data files can be obtained from the 'live' database or from a restored copy of the database.



Hours gathering repair data



Minutes making the repair

The Native Mode Repair by Key Value feature (option 2.5/4/3) is then invoked to read the data files and make a precise repair in the minimum amount of exclusive access time.

Minimizing database exclusive access repair time is a significant improvement in DBGeneral's database repair technology. The decoupling of gathering the repair data and making the repairs allows nearly continuous uptime of your production database. The diagnostics are preferably done on a copy of the damaged database in another account, or potentially on another machine (shadow?). The repair data is then applied to the production database with exclusive access. This technology has been specifically designed to allow decoupling with an eye toward maximizing production database uptime!

Detail Path Analysis performed on a copy of the database in a second account creates HFS file(s) indicating damaged keys.

```
Option 2.4 --- Analyze Performance of Detail Set Thu, Jul 13, 2000, 9:28 pm
SOURCE DATABASE: XYZDB.PUB.ACCT1
PASSWORD: ;
DATASET: IQ-HISTORY-D
Detail is pathed to the following sets:
Path # Search Key Related Master
-----
1 SERIAL-NUMBER SERIAL-NUMBER-A
2 ACCOUNT ACCOUNT-A
3 LAST-NAME LAST-NAME-A
Enter path number to analyze or '@' for all paths : 2
```

7.4 Option 2.4 'setup dialog'

Detail Dataset name (or number) is specified. Then a list of IMAGE paths are presented to which the user may respond '@' for all paths or with a specific path number.

```
NM Detail Diag 7.4.00(04/26/00) Copyright 1999, Bradmark, Inc.
Beginning Collection Phase... THU, JUL 13, 2000, 9:29 PM
---->Processing Chunk #01/03
10% complete. Cumul: 0:04:56 Remain: 0:44:25 Rate: 2363 Ent/sec.
/snip
100% complete. Cumul: 1:04:35 Remain: 0:00:00 Rate: 1806 Ent/sec.
End of data collection... THU, JUL 13, 2000, 10:33 PM
Active Entries: 6690961
Delete Chain Count: 300338
Wall time: 1:04:35 CPU time: 0:13:10.470

Organizing Data... THU, JUL 13, 2000, 10:33 PM
End of Data Organization... THU, JUL 13, 2000, 10:48 PM
Wall time: 0:14:13 CPU time: 0:13:42.289

Loading Chain Analysis Structure ... THU, JUL 13, 2000, 10:48 PM
10% complete. Cumul: 0:00:11 Remain: 0:01:44 Rate: 60270 Ent/sec.
/snip
100% complete. Cumul: 0:01:47 Remain: 0:00:00 Rate: 65034 Ent/sec.

End of analysis structure setup... THU, JUL 13, 2000, 10:50 PM
Wall time: 0:01:47 CPU time: 0:01:44.810
```

7.4 Option 2.4 'collection and organization phases'

These are the collection and organization phases. The dataset is read into an extract file of minimum size. Next, the data is organized to minimize Disc I/O during processing of the high-speed chain walk.

```
Validating Delete chain... THU, JUL 13, 2000, 10:50 PM
Delete Chain is valid.
End of Delete Chain analysis... THU, JUL 13, 2000, 10:50 PM
Wall time: 0:00:06 CPU time: 0:00:06.773
```

7.4 Option 2.4 'delete chain validation'

Here, the delete chain information has been made available and is validated.

```

-- Information for Detail Dataset IQ-HISTORY-D      8

Entry Length      591      Media Length      603
Blocking Factor   3        Block Size       1920
Number of Paths   3        Capacity         7000002

-- TurboImage/Serial read Comparison

-----|-----|-----|
Active Entries   | 6690961 | 6690961 |
High Water Mark | 6991299 | 6991299 |
Free Space Count| 309041  | 309041  |
Delete Chain Count| 300338  | 300338  |

```

Information for the detail dataset is presented as reported by IMAGE dbinfo intrinsics vs DBGeneral specific counts. Any discrepancies are reported as errors.

7.4 Option 2.4 'IMAGE vs DBGeneral counts'

```

Beginning Analysis Phase...

Examining detail chains... THU, JUL 13, 2000, 10:50 PM
Path 2 Item ACCOUNT Master set 2 - ACCOUNT-A
WARNING ! Stand-alone master entry, master record 2
Master Key: >4226310320339354 <
      (HEX): >3432 3236 3331 3033 3230 3333 3933 3534 2020 2020<
10% complete.  Cumul: 0:00:40  Remain: 0:06:07  Rate: 16368 Ent/sec.
20% complete.  Cumul: 0:01:20  Remain: 0:05:23  Rate: 16569 Ent/sec.
30% complete.  Cumul: 0:02:01  Remain: 0:04:42  Rate: 16589 Ent/sec.
40% complete.  Cumul: 0:02:54  Remain: 0:04:21  Rate: 15341 Ent/sec.
50% complete.  Cumul: 0:03:48  Remain: 0:03:48  Rate: 14630 Ent/sec.
60% complete.  Cumul: 0:04:29  Remain: 0:02:59  Rate: 14892 Ent/sec.
70% complete.  Cumul: 0:05:10  Remain: 0:02:12  Rate: 15099 Ent/sec.
ERROR ! Invalid master fwd chain pointer, master record 10140186
      Master Key: >11111111111111111111<
      (HEX): >6C6C 6C6C 6C6C 6C6C 6C6C 6C6C 6C6C 6C6C 6C6C 6C6C<
80% complete.  Cumul: 0:05:21  Remain: 0:02:01  Rate: 15094 Ent/sec.
90% complete.  Cumul: 0:06:36  Remain: 0:01:23  Rate: 13928 Ent/sec.
100% complete. Cumul: 0:08:04  Remain: 0:00:39  Rate: 12780 Ent/sec.

End of chain analysis... THU, JUL 13, 2000, 10:58 PM
Detail Chains Examined: 6690946
Master chain entries : 6501381
Chained masters      : 5914574
Stand-Alone Master Entries: 1
Detail chain tails  : 5914585
Detail chain heads  : 5914585
Wall time: 0:08:39 CPU time: 0:04:54.914

```

All preparations are complete, now DBGeneral does the high-speed chain walk and determines the validity of master chain heads, key values and detail inter-record pointers. Errors are displayed to include the detail or master record number and the key value in both ASCII and HEX formats. Summary numbers are reported.

7.4 Option 2.4 'Analysis Phase'

```

Searching for orphan entries... THU, JUL 13, 2000, 10:58 PM
ERROR ! Orphaned detail entry located at 337255
  Detail Key: >0741105430 <
      (HEX): >3037 3431 3130 3534 3330 2020 2020 2020 2020 2020<
ERROR ! Orphaned detail entry located at 420844
  Detail Key: >4227097382382419 <
      (HEX): >3432 3237 3039 3733 3832 3338 3234 3139 2020 2020<
/snip
ERROR ! Orphaned detail entry located at 4591407
  Detail Key: >267319 <
      (HEX): >3236 3733 3139 2020 2020 2020 2020 2020 2020 2020<
ERROR ! Orphaned detail entry located at 5573448
  Detail Key: >3207924420 <
      (HEX): >3332 3037 3932 3434 3230 2020 2020 2020 2020 2020<

15 Orphan Entries Found.

```

In order to completely analyze the detail, DBGeneral now looks at a structure to determine which entries in the detail have not been visited on the delete chain or in high-speed chain traversal and reports these as 'ORPHAN' entries. Their record number and key value are reported.

7.4 Option 2.4 'Orphan Entries'

```

Completed search for orphan entries... THU, JUL 13, 2000, 10:58 PM
Wall time: 0:00:00 CPU time: 0:00:00.87

Overall Processing Time:
Wall time: 1:29:31 CPU time: 0:33:39.522

```

7.4 Option 2.4 'Overall Processing Time'



Option 2.4 leaves behind HFS file(s) describing the bad key(s) for each damaged path.

```

:LISTFILE /ACCT1/PUB/XYZDB##.P##,2
PATH= /ACCT1/PUB/

CODE  -----LOGICAL RECORD-----  ---SPACE---  FILENAME
      SIZE  TYP      EOF      LIMIT R/B  SECTORS #X MX
      12W  FB          1      500000  1      256  1  *  XYZDB07.P02
      8W  FB          3      500000  1      256  1  *  XYZDB08.P01
      12W  FB         16      500000  1      256  1  *  XYZDB08.P02
      8W  FB          3      500000  1      256  1  *  XYZDB09.P01
      12W  FB          2      500000  1      256  1  *  XYZDB11.P02
  
```

HFS repair files shown with 'LISTFILE' command

For each of the paths with errors, DBGeneral leaves behind a HFS file containing the repair information. These files' naming convention is dataset filename appended with '.P##', where ## represents the path number. Remember this is HFS, so you must use the LISTFILE command.

Option 2.5/4/3, 'repair by key value', run on the production database, reads a copy of the HFS files and performs repairs from the list of damaged keys.

```

Option 2.5 --- Analyze & Correct Defective Paths  Fri, Jul 14, 2000,  4:05 am
SOURCE DATABASE: XYZDB.PUB.ACCT1
ENTER PASSWORD: ;

Disabling base for TPI during processing

DATASET: IQ-HISTORY-D

Entries in Set: 6690961
Capacity of Set: 7000002

The following methods are available :
1 .. Identify broken chains & defective keys on all paths (default)
2 .. Identify broken chains & defective keys starting with a specified path
3 .. Identify missing chain heads for a specified path
4 .. Repair broken chains and defective keys
Enter method requested (1:4) : 4

Do you have a current backup of this Database ? (N/Y) : Y

Dataset has no sort item from Master - SERIAL-NUMBER-A.
Do you wish to select one for chain reconstruction ? (N/Y) : N
Dataset has no sort item from Master - ACCOUNT-A.
Do you wish to select one for chain reconstruction ? (N/Y) : N
Dataset has no sort item from Master - LAST-NAME-A.
Do you wish to select one for chain reconstruction ? (N/Y) : N

The following stages are available :
1 .. Identify and repair broken chains and defective keys
   starting with a specified path (default)
2 .. Specify broken chains and defective keys by master key value
3 .. Repair broken chains and defective keys using information
   collected by a previous diagnostic run of option 2.4 or 2.5.
4 .. Resume suspended serial search for unchained detail entries
5 .. Rebuild all chains for a specified path
Enter Option number required (1:5) : 3

Utilize data from 2.4 Diagnostics for repair ? (Y/n/@) : @
Performing repairs for path 2, based upon 2.4 Diagnostics conducted
on FRI, JUL 14, 2000,  4:05 AM

Initiating High-Speed Key Repair ---
  
```

7.4 Option 2.5 repair from HFS files 'setup dialog'

Option 2.5 is run to repair the damage found by option 2.4. Database and dataset prompts are specified. Next 'repair broken chains and defective keys' is selected as method '4'.

If a sort path exists, DBG will sequence by the sort item. Otherwise, DBGeneral will prompt for an optional user specified 'sort' item.

Stage 3, "Repair broken chains and defective keys using information collected by a previous diagnostic run of option 2.4 or 2.5" is chosen. Then '@' is specified to direct option 2.5 to use 2.4 data for repairs on any path for this detail dataset. Repair data found is identified by path and creation date.

```

Native Mode Path Repair 7.4.00 (11/12/99)
Copyright (C) Bradmark Technologies, Inc. Houston TX, 1996

Sorting key selections (13 keys)...FRI, JUL 14, 2000,  4:05 AM
Sorting completed...FRI, JUL 14, 2000,  4:05 AM

Removing duplicate keys...FRI, JUL 14, 2000,  4:05 AM
0 Duplicates removed, 1 chain heads zeroed...FRI, JUL 14, 2000,  4:05 AM
  
```

7.4 Option 2.5 repair from HFS files 'Sort, Zero Chain heads'

The first steps in repair are to sort the bad key data and remove duplicates, then bad chain heads for which there are no details are zeroed.

```

Building path 2 of dataset 8 in database INQDB.PUB.PROD.
04:05:13 Gathering information and building work files...

04:10:14 Detail extract process (1/3): 561456 records processed
out of 3355338 total (16%) (still running)
04:10:14 Detail extract process (2/3): 565488 records processed
out of 3355338 total (16%) (still running)
/snip
04:42:52 Detail extract process (1/3): 3355338 records processed
out of 3355338 total (100%) (completed)
04:42:53 Detail extract process (2/3): 3327408 records processed
out of 3355338 total (99%) (still running)
04:43:19 Detail extract process (2/3): 3355338 records processed
out of 3355338 total (100%) (completed)
04:45:17 Detail extract process (3/3): 264096 records processed
out of 289326 total (91%) (still running)
04:45:36 Detail extract process (3/3): 289326 records processed
out of 289326 total (100%) (completed)

```

A sub-process handles the detail extract. For JUMBO datasets, there is a sub-process for each chunk which speeds throughput. By default each process reports every 5 minutes and when it completes. Here only the entries on the bad key list are being extracted.

7.4 Option 2.5 repair from HFS files 'Detail Extract'

```

04:45:36 Chain link process (1/5): 0 records processed
(still running)
04:45:37 Chain link process (1/5): 15 records processed
(completed)
04:45:39 Master update process (2/5): 0 records processed
(still running)
04:45:40 Detail update process (3/5): 9 records processed
(completed)
04:45:40 Master update process (2/5): 0 records processed
(still running)
04:45:43 Master update process (2/5): 12 records processed
(completed)
04:45:44 Detail update process (4/5): 6 records processed
(completed)
04:45:44 Detail update process (5/5): 0 records processed
(completed)
Detailed statistics for the path build:

Overall:
cpu time : 0:25:08.832
wall time: 0:40:31.302
num detail entries: 15
new master entries: 0
    primaries: 0
    secondaries: 0

Part 1 -- build work files:
cpu time : 0:24:59.726
wall time: 0:40:22.493
Part 2 -- update path:
cpu time : 0:00:06.971
wall time: 0:00:08.797
No b-tree index changes necessary.

Wall time : 0:40:31.302
CPU time : 0:25:08.832
Total detail entries : 15
Total master entries created : 0
    Primaries : 0
    Secondaries : 0

Completed processing of 2.4 Diagnostic repair data.
Base re-ENABLED for TPI

```

Chain link produces a list of inter-record detail pointer information, Master Update writes chain head data and Detail Update writes the inter-record detail pointer data. Notice here there are three processes doing the Detail Update since the JUMBO detail dataset has three chunks.

Overall processing data is reported. Here about 40 minutes to re-link 15 detail entries out of 6.7 million.

7.4 Option 2.5 repair from HFS files 'Chain Link, Master Update, Detail Update'

# Support For Changes in IMAGE

## Sixteen million+ block datasets supported in all options (80 gigabyte)

DBGeneral has been enhanced to correspond with the TurboIMAGE/XL enhancement, in version C.07.14, which will allow data sets to be up to 80 GB in size, twice as big as the old limit. This is due to changes in the IMAGE POINTER FORMAT block number from a signed value to an un-signed value. That extra bit doubles the number of possible blocks.

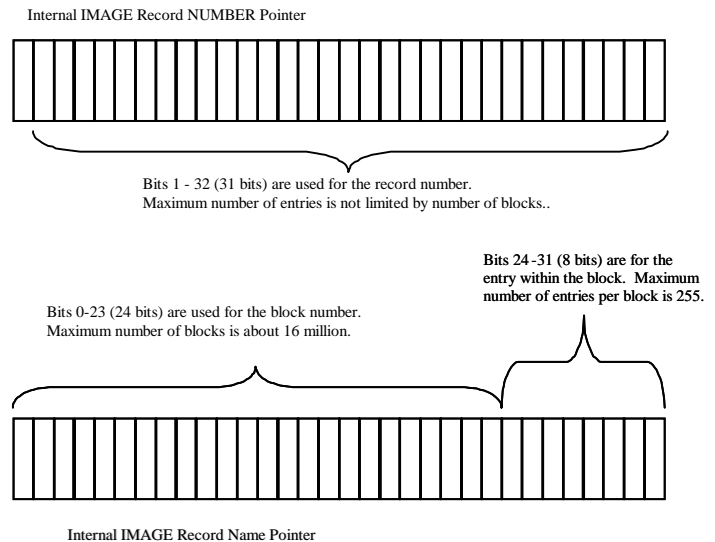
## IMAGE Expansions: Paths, Items, Sets, Pointer Format

Support for 64 paths on Master datasets

Support for Increased number of IMAGE Items

Support for Increased number of IMAGE datasets

Support for IMAGE Record Number format pointers



## Dialog Changes in DBGeneral

### Master Capacity Change (option 3.3) now prompts ‘Is this capacity correct’ in Batch

Master Capacity change’s (option 3.3) dialog in batch has been changed. Now the prompting sequence in batch is the same as on-line. Previously, there was a difference in that ‘Is this capacity correct?’ was not prompted in batch. A comparison of current and prior batch and online dialogs are shown in the table. Please adjust your JCL accordingly. This affects DBGENALT.cmd.vesoft. The new version is DBGENALT.MPEX.BRADMARK.

Current Batch Dialog	Current On-Line Dialog	Description	Prior Batch Dialog	Prior Online Dialog
3.3	3.3	Option	3.3	3.3
FINDB	FINDB	Basename	FINDB	FINDB
i	i	Password	i	i
Y	Y	Backup	Y	Y
IMMAS	IMMAS	Dataset	IMMAS	IMMAS
1000	1000	New Capacity	1000	1000
4	4	Option required: round up to prime, down to prime, sample, use specified	4	4
Y	Y	Capacity confirmation prompt “Is the new capacity of 1000 correct ? (Y/N) :”	Previous versions did not prompt this in batch.	Y

For JCL where the old dialog is desired, setting the JCW DBGUSERVERSION to 7217 “:SETJCW DBGUSERVERSION = 7217” will cause the dialog to continue as previous version of DBGeneral.

# DBGeneral New, Enhanced or Repaired Features Overview

## Improved Various Read Only Options to Not Change the MODIFY Date of Datasets

Previously, some options of DBGeneral such as 1.4 would change datasets' modify date. This has been corrected such that only access date is now updated.

## Option 1.4 , Jumbo Master, Dynamic Master Dataset Support

For master data sets that are JUMBO, DBGeneral displays a type of 'MJ'. Total current capacity is show as is capacity for each chunk.

Dataset	Typ	Current Capacity	Entry Count	Ent Lgt	Pth Cnt	Med Lgt	Blk Fct	(%) Ful	Volume Restrict
1 AUTO	MJ	307	10	15	1	26	30	3	DISC
	SIMP01.001	150							
	SIMP01.002	150							
	SIMP01.003	7							

7.4 Option 1.4 shows JUMBO chunks

For sets which are MDX (Master Dynamic Dataset eXpansion), DBGeneral displays a type of 'MX' and the MDX parameters.

MDX item	Description
Current Capacity	Total of original (or hash capacity) plus expansion area.
Maximum Capacity	The upper limit of capacity, beyond which the dataset will not expand
Increment Capacity	The number of entries by which the dataset will expand.
Hash Capacity	This is the capacity used in the hash algorithm of which capacity is a component. This is also the original area onto which expansion area is added.

Table of MDX items

Dataset	Typ	Current Capacity	Entry Count	Ent Lgt	Pth Cnt	Med Lgt	Blk Fct	(%) Ful	Volume Restrict
1 SET1	MX	3000	2968	273	0	278	1	49	DISC
		Maximum/Increment/Hash Capacity:	6000/1000/2000						
2 SET2	MX	11001	10615	9	0	14	19	96	DISC
		Maximum/Increment/Hash Capacity:	11001/1007/7011						
3 SET3	MX	2028	2021	2	0	7	39	100	DISC
		Maximum/Increment/Hash Capacity:	2028/507/1014						

7.4 Option 1.4 shows MDX parameters

## Option 1.6 Supports Lock Descriptors for Very Large Keys

Option 1.6 utilizes DBUTIL to integrate key information into useable format. This feature has been enhanced to support multiple line lock descriptors that occur when key values are very large.

## Option 2.1 and 2.2 Jumbo Master Support, Dynamic Master Set Capacities

```

--- General Information for set SET1

Set Number          1
Set Type            Dynamic Master
Entry Size          273
Record Size         278
Blocking Factor     1
No. of Paths        0

Hash Capacity       2000
Current Capacity    3000
Increment           1000
Maximum Capacity    6000
Active Entries      2968
High Water Mark     2971
    
```

7.4 Option 2.1 shows MDX parameters

## Option 2.4 False Error When Delete Chain Length Equals One

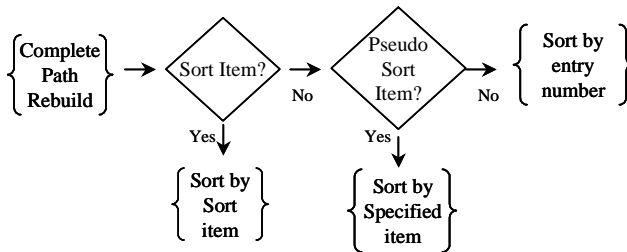
DBGeneral Option 2.4 has been repaired to correctly interrupt a delete chain with a length of one. Previously, option 2.4 would report an error when the dataset was actually structurally correct.

## Option 2.4 Single Path Analysis, 50% faster

See “Diagnostic Integration Providing Selective Key Value Repair” and “Faster Path Level Diagnostics “ in this document.

## Option 2.5/4/5 Complete Path Rebuild, Supports User-Selectable Sort Field

For paths without a sort item, you may now sequence chains for a dataset by any specified item.



```

Dataset has no sort item from Master - AUTO.
Do you wish to select one for chain reconstruction? (N/Y) : Y

ITEM NAME: ITEM1
    
```

7.4 Option 2.5 method 4 stage 5, prompts for high-speed sort item

## Option 2.6 Opens Database Exclusive by Default

Option 2.6 now opens the database in exclusive access by default. Previously the default was shared access. The two primary functions of the Global diagnostics are: performance analysis and structural analysis. Structural analysis may report false positives when running with other write access users of the database and therefore should be run either exclusive or with ONLY shared access users.

```
Option 2.6 --- Global Diagnostic ----- Thu, Aug 17, 2000, 10:03 am
SOURCE DATABASE: TECHS
PASSWORD: (N/A)
NOTE ! Exclusive access cannot be obtained because the following
        users are currently accessing the database :

HP30391C.07.10 TurboIMAGE/XL: DBUTIL (C) COPYRIGHT HEWLETT-PACKARD COMPANY 1987
>> For database TECHS
PIN PATH      EXECUTING PROGRAM      JOBNUM MODE
-----
75 1         QUERY.PUB.VESOFT          #S8      5
```

*7.4 option 2.6 open exclusive fails when other accessors*

In an effort to produce better structural analysis, the default is now exclusive access. This may be safely overridden when there are known to be only READ access users by specifying the DBOPEN mode after the database name:

```
Option 2.6 --- Global Diagnostic ----- Thu, Aug 17, 2000, 10:06 am
SOURCE DATABASE: TECHS,5
PASSWORD: (N/A)
NOTE ! DBOPEN mode changed from 3 to 5
```

*7.4 option 2.6 open exclusive override to SHARED mode 5*

## Option 2.6 Computing Environment Information and \$stdlist enhancements

Option 2.6 has been enhanced with a “Computing Environment Information” section to define the machine characteristics on which the global has been executed. Run time is displayed at the end of the report.

```
#####
##/##/####          COMPUTING ENVIRONMENT INFORMATION          Page 1
                      (DBX.WORK2.ATDBG08)
=====
CPU Model           : SERIES 918LX          HPSUSAN           : 1136156341
MPE/iX Version      : C.70.00              IMAGE/SQL Version : C.10.01
DBGeneral Version   : 7.4.11
=====
```

*7.4 option 2.6 Computing Environment Information*

Also, the \$stdlist will now display the structural status of the database and run time.

```
Database is Structurally Sound.
Wall Time: 00:01:40; CPU Time: 00:00:07.589
```

*7.4 option 2.6 \$stdlist structural status and run time*

## Option 3.3 IMAGE Root File Version Update

When a database has a single Btree, IMAGE DBCONTROL mode 13 fails to correctly update the root file version when adding a Btree. This has been worked around such that DBGeneral now updates the root file version.

### Option 3.3 IMAGE Root logical\_eof correction (DBUTIL>>Erase)

When using DBUTIL to erase a database, you may experience an error due to previous versions of DBGENERAL not correctly setting the logical\_eof in the dataset control block area of the root file. Only DBUTIL>> erase seems to use the data. The logical\_eof is not correctly set.

### Option 3.5 detail dataset 80 gig support

See "Support For Changes in IMAGE" in this document

Prior versions of DBGeneral might give errors that the calculated size (number of blocks) for the dataset were exceeded. With the addition of support for 80 Gigabyte datasets, this has been corrected.

### Option 3.5 Compatibility with ONLINE BACKUP jcw DBG35COPY

By default option 3.5 uses an expand in place method which speeds the expansion of detail datasets but may be in conflict with some ONLINE BACKUP file label flags. To permit detail capacity changes which are compatible with ONLINE BACKUP, DBGENERAL provides an option which is enable with the JCW DBG35COPY = 1.

With the JCW set to 1, DBGENERAL will build a new dataset and COPY the data from the old dataset. This method is slower, but compatible with ONLINE BACKUP. See DBGENALT which will automatically enable this option during processing as needed.

```
Capacity change in progress ---
Database access disabled.
Note: Utilizing 'COPY' technique for capacity change.
Dataset change successfully completed
```

*7.4 Option 3.5, DBG35COPY JCW enables copy*

### Option 3.6 Choice of path during Reorganize (repack) detail dataset

```
Detail D16 is pathed to the following sets:
  Path #      Related Master
    1         M1
    2         M2
/snip
   15        M15
   16        M16          <= Primary path

There are four detail set reorganization methods available:
/snip
  4 .. Method 4 :
      Same as method 1,
      but allows the user to select which path
      is to be used to organize the dataset.
      Which method do you wish to use ? (1:4) : 4

Enter path to use for reorganization (1:16) : 9
Reorganize dataset along path 9, correct? (Y/n) : Y

/snip
Using method 4
NM Reorg module 7.4.00 (05/01/00)
Detail set : D16          Entries: 4465
Number of paths: 16
The ! denotes the selected path.
Master set:   1  Entries: 31
/snip
Master set: !  9  Entries: 31
Master set:  10 Entries: 31
/snip
Master set:  16 Entries: 38
Preparation phase is completed.
Database access is now disabled.
Phase 1 - Reorg will now begin along the selected path.
WED, AUG  2, 2000, 1:57 PM
```

*7.4 Option 3.6 method 4, permits selection of path*

## **Option 4.4 Erase Dataset Manages MDX and DDX**

Option 4.4 now returns dynamic datasets to their pre-expansion disc allocation. MDX datasets that are erased will have their expansion area removed and DDX datasets will have their current capacity returned to initial capacity. If there have been expansions, this will result in a savings of disc space. Dataset characteristics are reset accordingly: IMAGE current capacity is reset to initial / hash capacity.

## **Option 4.6 Copy of Base Open for Read Access Avoids DBOPEN Errors**

When a database is open for READ access, DBGeneral option 4.6 will allow it to be copied. In some cases this would produce a copied database for which DBOPEN would fail `DBOPEN fails MPE FILE ERROR -247 RETURNED BY FOPEN ON DBG`. This problem has been resolved with an adjustment to the copied database's root file.

## **Option 4.7 permits direct specification of blocking factor**

When a dataset is re-blocked the new dataset's characteristics may be specified as block size or as blocking factor. Previously, only block size was permitted.

## **Option 5.# Permits Placement Before or After an Entity**

When a database is structurally changed via options 5.2 thru 5.5, the target item or set being added or moved is specified relative to an item in the item list, an item in the set-item list or a set in the set list. This relative entity may now be specified as PRIOR in addition to the default. For example, the location in the set list must be specified as before or after an existing dataset when adding a new dataset. The specification of a '-' minus sign prior to the entity will cause the item or set to be located 'prior' to that entity. Previously, only 'after' was permitted. After remains the default for specifications where the minus sign is not specified.

## **Option 8.1 DB Key Change Supports Critical Item Update (CIUpdate)**

DB Key Change has been enhanced to utilize the IMAGE Critical Item Update feature. This produces enhanced performance by substituting DBUPDATE for DBDELETE / DBPUT. The database must have Critical Item Update set ON or ALLOWED.

## **Improved Third Party Indexing processing for OMNIDEX**

Option 3.6 calls TPIMAIN to process TPI Indexes on the dataset reorganized. Previously the text output to \$stdlist for OMNIDEX TPI Indexing was suppressed. This process has been enhanced to display the dialog from TPIMAIN's call to OMNIDEX. An option has been included as a JCW named DBGTPIOUTPUT such that non-zero values will suppress the dialog. Further, it is noted that when the OMNIDEX product is installed in an account other than DISC a variable DISC\_ACCOUNT should be set system wide (or at least prior to DBGeneral's invocation) to a string value of the account name where the DISC product is installed.

For example,  
`:setvar disc_account = "disc304"`  
would be appropriate when utilizing the disc304 account.

## **Option 3.2, 3.3, 3.4 and 3.5 do NOT to disable for TPI during processing.**

Previously, DBGENERAL disabled TPI indexing during capacity changes. Currently, DBGENERAL does not require indexing to be disabled and these options no longer disable indexing at any time.

## **MPE/iX 7.0, A-class and N-class e3000 computers**

DBGENERAL 7.3.15 and later has been certified with MPE/iX 7.0 running on A and N class e3000 computers.